# This paper is a Post-Print version (i.e. final draft post-refereeing). 

 The publisher's version can be accessed through DOI: (to be announced)
# A Reversible Dynamic Movement Primitive formulation 

Antonis Sidiropoulos ${ }^{1}$ and Zoe Doulgeri ${ }^{2}$


#### Abstract

In this work, a novel Dynamic Movement Primitive (DMP) formulation is proposed which supports reversibility, i.e. backwards reproduction of a learned trajectory. Apart from sharing all favourable properties of the original DMP, decoupling the teaching of position and velocity profiles and bidirectional drivability along the encoded path are also supported. Original DMP have been extensively used for encoding and reproducing a desired motion pattern in several robotic applications. However, they lack reversibility, which is a useful and expedient property that can be leveraged in many scenarios. The proposed formulation is analyzed theoretically and its practical usefulness is showcased in an assembly by insertion experimental scenario.


## I. Introduction

The technological advancements in the field of robotics over the past few years have triggered a surge in research interest for incorporating robots more and more actively in everyday life. However, programming a robot to perform even simple tasks usually requires considerable amount of time and advanced robotic and programming knowledge. To facilitate this cause, the use of Programming by Demonstration ( PbD ) has been proposed for passing on human skills to a robot [1]. At the core of this approach is a model used for encoding the demonstrated trajectory encapsulating the desired motion pattern for performing a task. The properties of this model play a major role, as they determine whether the encoded motion is amenable to generalization, online adaption, reverse execution etc, depending on the target application.

In the literature there are a lot of approaches for encoding a desired motion pattern, such as spline decomposition [2], neural networks [3], Hidden Markov Models [4], Gaussian Process Regression [5], Gaussian Mixture Models [6] and more. Undoubtedly, one of the most prominent approaches is the Dynamic Movement Primitives which have been applied in numerous robotic applications [7]-[10]. The reason behind this widespread applicability of DMP is mainly attributed to the favourable properties they come along with. Specifically, they offer a fairly simple framework for encoding any smooth non-linear trajectory, allowing generalization to new targets (end points) and changing the speed of execution. Moreover, they guarantee global asymptotic stability (GAS) at the target, robustness to target perturbations and allow seamlessly the encoding and synchronization of multiple

[^0]DoFs. All these properties make DMP quite the appealing and efficient choice to opt for when encoding a desired motion pattern. However, there is still room and need for improvement as original DMP do not support reversibility. There are many tasks where reversibility, i.e. reproduction of the learned trajectory backwards in time, could come quite in handy.

For many tasks, reverse execution could enable automatic derivation of certain required operations from their forwards counterparts. For instance, in assembly tasks a reversible model could be used for assembling and disassembling a part. In general, when performing any task that involves reaching a target, like a handover, a placing of a part etc., the reverse motion could be used for the retraction, saving the effort of explicitly encoding and programming the backward motion. This becomes even more useful in tasks executed in cluttered environments. Reversibility can also prove really effective and practical for recovering from errors during the execution of a task. These errors could originate from faulty or noisy sensor measurements or could be attributed to disturbances. Using reverse execution, the robot can temporarily back out of an erroneous situation by tracing back to a previous point, after which the execution can be automatically retried. There are some works that have applied reversibility to recover from such errors in practical applications [11], [12]. In these works, the authors address error recovery by reversing the program's execution by means of designing a domain specific language (DSL) which supports running backwards the executed programming commands. Reversibility using DMP is addressed in [13] with a DMP formulation which however fails to ensure global asymptotic stability (GAS) and reversibility. Eventually the authors resort to the use of two separate DMPs to ensure both. Two separate DMPs to achieve reversibility are also used in [14].

In this work, we propose a novel DMP formulation that achieves reversibility as opposed to the original DMP, maintaining all its favourable properties. Moreover, it allows a two phase PbD for velocity profile teaching. Learning the velocity profile in a second phase is particularly suitable for encoding tasks that require high precision during the demonstration of the desired path. The proposed formulation allows not only replaying the trajectory back from its goal, but also moving back and forth along the learned path anywhere during the execution. Other qualities of the proposed formulation is that training requires only position measurements and is decoupled from the DMP's stiffness and damping gains. Additionally, the DMP's effective stiffness and damping is not dependent on the temporal scaling parameter. The
proposed DMP is theoretically analyzed and compared with the original one. Its benefits are demonstrated experimentally in an assembly by insertion scenario inspired by a real case.

The rest of this paper is organized as follows: section II provides an overview on original DMP. In section III the proposed DMP formulation is presented and its properties are detailed in section IV. Experimental results are presented in section V and conclusions are drawn in section VI. In the appendix, the correspondence between the proposed and original DMP formulations is analyzed and preliminaries on unit quaternions are provided.

## II. Dynamic Movement Primitives Preliminaries

DMP consist of a transformation system which generates a trajectory and a canonical system for controlling the system's temporal evolution [15]. The transformation system is composed of a second order linear attractor to a goal and a non-linear forcing term which has to be learned in order to encode a desired kinematic behavior. Aggregating the different variants that have been proposed in the literature in a more generic formulation a 1-DoF DMP is given as follows:

$$
\begin{align*}
& \tau^{2} \ddot{y}=\alpha_{z} \beta_{z}(g-y)-\alpha_{z} \tau \dot{y}+g_{f}(x)\left(g-y_{0}\right) f_{s}(x)  \tag{1}\\
& \tau \dot{x}=h(x) \tag{2}
\end{align*}
$$

or expressing (1) in state equations introducing $z=\tau \dot{y}$ :

$$
\begin{align*}
& \tau \dot{z}=\alpha_{z} \beta_{z}(g-y)-\alpha_{z} z+g_{f}(x)\left(g-y_{0}\right) f_{s}(x)  \tag{3}\\
& \tau \dot{y}=z \tag{4}
\end{align*}
$$

where $y, \dot{y}$ is the position and velocity, $g$ the target, $y_{0}$ the initial position. The phase variable $x$ is used to avoid direct dependency on time and $\tau>0$ is a temporal scaling parameter, typically set equal to the movement's total duration $T=t_{f}-t_{0}$, where $t_{0}$ is the initial and $t_{f}$ the final time instant. The forcing term $f_{s}(x)$ is given by:

$$
\begin{equation*}
f_{s}(x)=\boldsymbol{\phi}(x)^{T} \boldsymbol{w} \tag{5}
\end{equation*}
$$

where $\boldsymbol{\phi}(x)^{T}=\left[\begin{array}{lll}\psi_{1}(x) & \cdots & \psi_{N}(x)\end{array}\right] / \sum_{i=1}^{N} \psi_{i}(x)$, with $\psi_{i}(x)=\exp \left(-h_{i}\left(x-c_{i}\right)^{2}\right)$. The gating function $g_{f}(x)$ ensures that the forcing term eventually vanishes thus (1) acts as a spring-damper and converges asymptotically to the goal $g$.

The canonical system's evolution is determined by (2), with $h(x)$ chosen so that $x$ evolves monotonically from its initial value $x(0)=x_{0}$ to its final value $x\left(t_{f}\right)=x_{f}$. Many options are available, e.g. a linear canonical system $\tau \dot{x}=1$ for $x \leq 1$ and $\tau \dot{x}=0$ otherwise, with $x_{0}=0$ and $x_{f}=1$; an alternative choice is an exponential one $\tau \dot{x}=-a_{x} x$ with $x_{0}=1$ and $x_{f}=0^{+}$. For the gating function $g_{f}(x)$ there are many choices as well, such as linear, exponential or sigmoid gating, to name but a few. Regarding the choice of $\alpha_{z}, \beta_{z}$ a typical approach is to set $\beta_{z}=\alpha_{z} / 4>0$ to render the linear part of (1) critically damped. To achieve good approximation during learning, a general heuristic is to place the centers $c_{i}$ of the Gaussian kernels in (5) equally spaced in time. A typical choice is to then set the inverse widths of the

Gaussians as $h_{i}=\frac{a_{h}}{\left(c_{i+1}-c_{i}\right)^{2}}, h_{N}=h_{N-1}, i=1, \cdots, N$, where $a_{h}>0$ is a scaling factor controlling the overlapping between the kernels.

## III. A Reversible Dynamic Movement Primitive

Reversibility of DMP means that the generated system's trajectory will be identical in forward and backward motion. Given a goal attractor dynamics e.g. (1), it is however impossible to have global stability and reversibility simultaneously as also noted in [13]. Original DMP are indeed not reversible as shown at the end of Appendix A. In order to produce a reversible DMP the authors in [13] suggest a non-linear system with two equilibria, an attractor and a repeller, and build their proposed formulation around it. This formulation achieves partial reversibility if the solution stays within a region and becomes unstable beyond. As the solution cannot be guaranteed to stay within the permitted region, which is expected as the basic dynamics are non-linear, the authors finally resort to using two forcing terms to ensure global stability. Instead of having a goal attractor as a foundation of the transformation system we base our proposed formulation on a linear system with a global asymptotically stable origin. Inspired by linear trajectory tracking dynamics we propose the following DMP structure for 1-DoF:

$$
\begin{align*}
& \ddot{y}=\ddot{y}_{x}-D\left(\dot{y}-\dot{y}_{x}\right)-K\left(y-y_{x}\right)  \tag{6}\\
& \dot{x}=h(x) / \tau, x\left(t_{0} / \tau\right)=x_{0}, x\left(t_{f} / \tau\right)=x_{f} \tag{7}
\end{align*}
$$

where $K, D$ are positive scalars, $h(x)$ evolves monotonically from $x_{0}$ to $x_{f}$ and must be zero for $x$ outside of the interval defined by $x_{0}$ and $x_{f}$ (the canonical system examples presented in Section II are appropriate choices); lastly, $y_{x}$ is the desired spatially and temporally scaled trajectory, generated by:

$$
\begin{align*}
& y_{x} \triangleq k_{s}\left(f_{p}(x)-f_{p}\left(x_{0}\right)\right)+y_{0}  \tag{8}\\
& \dot{y}_{x}(t)=k_{s} \dot{f}_{p}(x)  \tag{9}\\
& \ddot{y}_{x}(t)=k_{s} \ddot{f}_{p}(x) \tag{10}
\end{align*}
$$

where

$$
\begin{equation*}
k_{s} \triangleq \frac{g-y_{0}}{f_{p}\left(x_{f}\right)-f_{p}\left(x_{0}\right)} \tag{11}
\end{equation*}
$$

is the spatial scaling term and

$$
\begin{equation*}
f_{p}(x)=\phi(x)^{T} \boldsymbol{w} \tag{12}
\end{equation*}
$$

encodes the demonstrated position trajectory through a weighted sum of Gaussians, with $w=\operatorname{argmin}_{w} J\left(f_{p}(x), y_{d}\right)$, where $J$ can be any objective cost function, e.g. the Least Squares (LS) or the Locally Weighted Regression (LWR) objective [15]. The derivatives of $f_{p}(x)$ can be calculated analytically from $\dot{f}_{p}(x)=\left(\frac{\partial \phi}{\partial x} \dot{x}\right)^{T} \boldsymbol{w}$ and $\ddot{f}_{p}(x)=\left(\frac{\partial^{2} \phi}{\partial x^{2}} \dot{x}^{2}+\right.$ $\left.\frac{\partial \phi}{\partial x} \ddot{x}\right)^{T} \boldsymbol{w}$.

Forward reproduction of the learned trajectory is accomplished integrating (6)-(7). For backward reproduction only the sign of $\dot{x}$ needs to be flipped, i.e. $\dot{x}=-h(x) / \tau$, $x\left(t_{0} / \tau\right)=x_{f}$.

Denoting $e=y-y_{x}$ and considering that $D, K>0$ it follows from (6) that $e, \dot{e}, \ddot{e} \rightarrow 0$. What is more, as $y_{x}$ and its
derivatives are bounded by construction, $y, \dot{y}, \ddot{y}$ are bounded as well. In forward reproduction, (7) implies that $x \rightarrow x_{f}$ and $\dot{x}, \ddot{x} \rightarrow 0$, and from (8)-(10) and (11) we can conclude that $y_{x} \rightarrow g, \dot{y}_{x}, \ddot{y}_{x} \rightarrow 0$. Therefore, $y \rightarrow g, \dot{y}, \ddot{y} \rightarrow 0$. In backward reproduction from any $x \neq x_{0}, \dot{x}=-h(x) / \tau$ implies $x \rightarrow x_{0}$ and $\dot{x}, \ddot{x} \rightarrow 0$, which from (8)-(10) and (11) implies that $y_{x} \rightarrow y_{0}, \dot{y}_{x}, \ddot{y}_{x} \rightarrow 0$, hence $y \rightarrow y_{0}, \dot{y}, \ddot{y} \rightarrow 0$.

Notice that (6) can also be written as $\ddot{y}=K(g-y)-$ $D \dot{y}+f^{\prime}(x)$, where $f^{\prime}(x)=\ddot{y}_{x}+D \dot{y}_{x}-K\left(g-y_{x}\right)$, which is a linear system modulated by the nonlinear forcing term $f^{\prime}(x)$ which fades to zero as $t \rightarrow \infty$ and consequently it has a GAS point attractor like the original DMP.

For Cartesian position encoding the above formulation is readily extended as each DoF can be encoded separately and synchronization can be achieved using the same canonical system (7) for all DoFs. Regarding Cartesian orientation, a formulation analogous to (6) is adopted, which makes use of the quaternion logarithm as in [16] (a brief summary on unit quaternions is provided in Appendix B). Introducing $\boldsymbol{\eta} \triangleq \log \left(\boldsymbol{Q} * \overline{\boldsymbol{Q}}_{0}\right)$, where $\boldsymbol{Q}$ is the current, $\boldsymbol{Q}_{0}$ the initial orientation expressed as unit quaternions, $*$ denotes the quaternion product, (.) the quaternion inverse and $\log ($. the quaternion logarithm, the proposed DMP formulation for Cartesian orientation is given by:

$$
\begin{equation*}
\ddot{\boldsymbol{\eta}}=\ddot{\boldsymbol{\eta}}_{x}-\boldsymbol{D}\left(\dot{\boldsymbol{\eta}}-\dot{\boldsymbol{\eta}}_{x}\right)-\boldsymbol{K}\left(\boldsymbol{\eta}-\boldsymbol{\eta}_{x}\right) \tag{13}
\end{equation*}
$$

with the canonical system given by (7). The matrices $\boldsymbol{K}, \boldsymbol{D}$ are diagonal positive definite stiffness and damping matrices. The desired spatially and temporally scaled trajectory is given by:

$$
\begin{equation*}
\boldsymbol{\eta}_{x} \triangleq \boldsymbol{K}_{s} \boldsymbol{f}_{q}(x) \tag{14}
\end{equation*}
$$

where $\boldsymbol{K}_{s} \triangleq \operatorname{diag}\left(\boldsymbol{\eta}_{g} \cdot / \boldsymbol{f}_{q}\left(x_{f}\right)\right)$ with $\boldsymbol{\eta}_{g} \triangleq \log \left(\boldsymbol{Q}_{g} * \overline{\boldsymbol{Q}}_{0}\right)$, $\boldsymbol{Q}_{g}$ is the target orientation and ./ denotes the element-wise division. The desired motion is learned based only on the demonstrated orientation $\boldsymbol{\eta}_{d}=\log \left(\boldsymbol{Q}_{d} * \overline{\boldsymbol{Q}}_{d, 0}\right)$, where $\boldsymbol{Q}_{d}$ and $\boldsymbol{Q}_{d, 0}$ are the demonstrated and initial orientation, through $\boldsymbol{f}_{q}(x)$ which is a weighted sum of Gaussians:

$$
\begin{equation*}
\boldsymbol{f}_{q}(x)=\boldsymbol{W} \boldsymbol{\phi}(x) \tag{15}
\end{equation*}
$$

where $\boldsymbol{W}=\left[\begin{array}{lll}\boldsymbol{w}_{x} & \boldsymbol{w}_{y} & \boldsymbol{w}_{z}\end{array}\right]^{T}$, with $\boldsymbol{w}_{i}$ being the weights for each coordinate $i \in\{x, y, z\}$. The desired scaled velocity $\dot{\boldsymbol{\eta}}_{x}$ and acceleration $\ddot{\boldsymbol{\eta}}_{x}$ are obtained by differentiating $\boldsymbol{\eta}_{x}(t)$ as in (9), (10). The orientation $\boldsymbol{Q}$, rotational velocity, $\boldsymbol{\omega}$ and acceleration $\dot{\omega}$ can be obtained from $\boldsymbol{\eta}, \dot{\boldsymbol{\eta}}, \ddot{\boldsymbol{\eta}}$ using $\boldsymbol{Q}=\exp (\boldsymbol{\eta}) * \boldsymbol{Q}_{0}$ and (30), (31) from Appendix B. Finally, the stabilty analysis for $1-\mathrm{DoF}$ is readily extended to the orientation formulation (13), as each DoF in $\boldsymbol{\eta}$ is decoupled.

## IV. Proposed DMP Properties

## A. Standard DMP properties

The proposed DMP formulation retains all desirable properties of the original DMP. Global asymptotic stability at the target has already been discussed in the previous section. It is also clear that the stable coordination of multiple DoFs is similar to the original DMP, using a common canonical system for all DoFs.

Spatial and temporal scaling of the novel DMP, i.e. generation of trajectories that are qualitatively similar or topologically equivalent [15] when there is a change in the initial/target position or the temporal scaling parameter $\tau$, follows straightforwardly from the mathematical correspondence between the two as detailed in Appendix A. It can also be easily verified through simulations that for different initial/target poses or temporal scaling, the trajectories produced by the proposed DMP coincide with those of the original one.
Robustness to perturbations is also a trait of the proposed formulation. Such perturbations could be for instance an external disturbance. In this case, as in the original DMP, the phase stopping mechanism is employed, which results in the slow down or even halt of the trajectory generation by modifying the canonical system's evolution. Denoting the disturbance by $d(t)$, phase stopping can be implemented by modifying the canonical system's evolution as follows:

$$
\begin{equation*}
\tau \dot{x}=\frac{h(x)}{1+a_{d}|d(t)|} \tag{16}
\end{equation*}
$$

with $a_{d}>0$. Other phase stopping types are also possible, like a sigmoid stopping [17]. Another type of perturbation is the change of the target position on the fly. The DMP will modulate its trajectory to reach the new target.

Another appealing property of DMP is the capability of incorporating coupling terms, so as to modify online the dynamical system's trajectory based on external signals, obviating the need of trajectory replanning. Coupling terms have been applied to adjust online the DMP's trajectory based on the external force measurements [7], to enforce position/joint limits [18], for obstacle avoidance [19] and other. These coupling terms, can also be included in the proposed DMP, to attain the desired behaviour. For limit avoidance, we can introduce the state $z=\dot{y}$, to rewrite the novel DMP in the form of state equations and then add the repulsive force at the velocity level as in the original DMP:

$$
\begin{aligned}
\dot{\boldsymbol{y}} & =\boldsymbol{z}-\frac{\gamma}{\left(\boldsymbol{y}_{L}-\boldsymbol{y}\right)^{3}} \\
\dot{\boldsymbol{z}} & =\ddot{\boldsymbol{y}}_{x}-\boldsymbol{D}\left(\boldsymbol{z}-\dot{\boldsymbol{y}}_{x}\right)-\boldsymbol{K}\left(\boldsymbol{y}-\boldsymbol{y}_{x}\right)
\end{aligned}
$$

where $\boldsymbol{y}_{L}$ is the limit and $\gamma>0$ controls the effect of the repulsive force. Obstacle avoidance can also be achieved as in [19]. Denoting $\boldsymbol{p}_{o}$ the obstacle's position, the term $\boldsymbol{f}_{o}=$ $\gamma \boldsymbol{R} \dot{\boldsymbol{y}} \phi e^{-\beta \phi}$ can be added to the DMP's acceleration, where $\gamma, \beta>0, \boldsymbol{R}$ is a rotation around the axis $\boldsymbol{k}=\left(\boldsymbol{p}_{o}-\boldsymbol{y}\right) \times \dot{\boldsymbol{y}}$ by angle $\pi / 2$ and $\phi=\cos ^{-1}\left(\frac{\left(\boldsymbol{p}_{o}-\boldsymbol{y}\right)^{T} \dot{\boldsymbol{y}}}{\mid\left(\boldsymbol{p}_{o}-\boldsymbol{y}\right)\|\dot{\boldsymbol{y}}\|}\right)$.

## B. Additional Properties

Apart from reversibility, the proposed formulation decouples the DMP's effective stiffness and damping from the temporal scaling parameter $\tau$ (see (6) compared to (22) for the original one), which can affect the system's response in the presence of perturbations or other coupling terms. Moreover, training in the novel DMP requires only position measurement, whereas the original DMP requires additionally velocity and acceleration measurements, which
are usually noisy affecting the accuracy of the learning process. Hence, computational load and memory resource demand is reduced with the proposed formulation. Apart from those advantages, two additional properties detailed below are supported.

## Decoupled teaching of path and velocity profile:

Demonstrating a desired trajectory using kinesthetic guidance can prove to be quite cumbersome, as one has to pay attention to guide the robot accurately along the desired path, while at the same time imposing the desired speed of execution (velocity profile). This places a lot of cognitive load to the user and can deteriorate the quality of the demonstration. Moreover, there are tasks where accuracy in the demonstrated path is of great essence. A two phase learning approach was initially presented in [20], which uses the original DMP and learns the velocity profile and stiffness in the second phase. With the proposed DMP, velocity profile teaching can also be utilized in a second phase and is easily realized, with a slight modification of the canonical system. In particular, after the careful demonstration of the path (phase 1), in phase 2 the robot is under position control, following the output produced by (6) which generates the path demonstrated during phase 1 , but with different speed according to the following modified canonical system:

$$
\left\{\begin{array}{l}
\ddot{x}=-d_{x} \dot{x}+f_{v}, \text { for } x<1  \tag{17}\\
\ddot{x}=\dot{x}=0, \text { for } x \geq 1
\end{array}\right.
$$

with $x(0)=1, \dot{x}(0)=0$, where $d_{x}>0, f_{v}=\boldsymbol{n}^{T} \boldsymbol{f}_{\text {ext }}$ with $\boldsymbol{n}=\frac{\partial \boldsymbol{y} / \partial x}{\partial \boldsymbol{y} / \partial x}$ being the unit vector pointing along the direction of the motion, $\boldsymbol{y}$ denotes the Cartesian position and $f_{\text {ext }}$ is the external force provided by the robot's F/T sensor. At the end of this process, the recorded robot's Cartesian pose is used to retrain the DMP.

## Bidirectional drivability along the path:

A similar modification of the canonical system can also be used for phase stopping and bidirectional drivability along the path:

$$
\begin{equation*}
\ddot{x}=-d_{x}\left(\dot{x}-\dot{x}_{d}\right)+f_{v} \tag{18}
\end{equation*}
$$

where $\dot{x}_{d}=s \frac{1}{\tau\left(1+a_{d}\left\|\boldsymbol{F}_{\text {ext }}\right\|\right)}$ with $x(0)=0, s=1$ for forward and $s=-1$ for reverse execution. Essentially, (18) is obtained by combining (16) and (17). In the absence of external disturbances, it follows that $\dot{x} \rightarrow s / \tau$ and the motion evolves autonomously. When external forces are applied, phase stopping ensures that $\dot{x}_{d} \rightarrow 0$ and the motion evolves according the force along the path $f_{v}$. Bidirectional drivability along the path can prove useful in many scenarios, e.g. to allow manual inspection of the executed trajectory, driving the robot back/forth along the learned path.

## V. Experimental Results

In this section we demonstrate the usefulness of the proposed DMP in a ring-in-hole ( RiH ) assembly. This assembly is inspired by a real case of a car starter assembly. The task involves the insertion of sliding rings with flexible contact wings shown in Fig. 1 into pallets consisting of a series of holes. We use the actual sliding rings and 3D printed pallets,
from the CAD model of the actual pallets (Fig. 1). The high flexibility and elasticity of the contact wings adds to the difficulty of the task as grasping the ring from the table and inserting it with the correct orientation is important because the contact 'wings' of the ring should be properly aligned with the side walls of the pallet for the subsequent moulding operation to be successful.


Fig. 1: Ring in hole.


Fig. 2: Ring in hole scene.

Based on this real case scenario, we have devised the task shown in Fig. 2 for the insertion of 2 rings into corresponding pallets. We use a UR5e robot, which provides measurements of external wrenches, and an RG2FT gripper attached at its wrist.

During teaching, the geometric constraints due to the pallet's shape and the tight boundaries of the hole make it difficult for a human to demonstrate the insertion as he/she may have to wiggle the object before it fits properly in the hole and the sockets. This however deteriorates the quality of the demonstration. We resolve this problem by demonstrating the disassembly trajectory (shown in the video), starting with the ring already properly aligned and inserted into a pallet. By flipping the collected data the encoding of the forward trajectory is then performed. The second phase of the training procedure (section IV-B) is then utilized to learn the velocity profile (shown in the video). Notice that in order for the DMP to generalize the learned motion pattern properly to new targets (pallet poses), the DMP is encoded w.r.t. the target pose (pallet pose). Further notice that by demonstrating the reverse, the retraction motion will respect the geometric constraints.

The trained DMP is used for executing all the required motions of the task in sequence, starting from the initial robot pose, shown in Fig. 2, and moving to the pick/ insertion target poses. Notice that all poses in Fig. 2 are different from
those employed during the demonstration. When the target is reached, the gripper closes (opens) for pick (insertion) and the robot retracts to its initial pose along the same (reverse) path. The latter is particularly important in the cluttered environment of the real case. To control the phase variable we use (18), applying also a dead-zone of $10 N$ to $\boldsymbol{F}_{\text {ext }}$ when the distance to the target is below 1.5 cm so that the robot can apply the required inserting force without stopping prematurely by the 'phase-stopping' mechanism.

We showcase the usefulness of the reversibility and bidirectional drivability (18) of the proposed DMP formulation in the context of manual trajectory inspection and automatic error recovery from faulty sensor measurements, exploring two specific scenarios. Specifically, during the execution of the first insertion, the user intervenes by driving the robot back/forth along the path to inspect the learned trajectory near the target. During the second insertion, a purposely wrong target is provided, which can be hypothetically attributed to perception errors of a vision sensor. This results in a collision with a surface generating a contact force and a premature stop, due to (18). Then we signal a partial retraction, utilizing (18) for reverse execution until the phase variable $x$ reaches 0.2 , and then provide the correct target pose and resume the insertion, restoring $s$ to its initial value. Notice that the detection for triggering this procedure could be automated by a higher level perception system, leveraging vision and/or force and proximity sensors. However, such detection schemes are beyond the scope of this work.


Fig. 3: Ring 1 insertion and retraction: Position and orientation trajectories along $z$-axis with user inspection.

Results are presented in Fig. 3-6. During the insertion of 1st ring the user intervenes (Fig. 3) to inspect the trajectory driving the robot back/forth along the path at $t=3.8-28 \mathrm{sec}$. During the 2nd ring insertion in Fig. 4, a collision occurs at $t=8.9$ and the robot stops as it can be seen from the phase variable $x$ and the response along the $z$-direction followed by a retraction until $x=0.2$ at $t=11.6$, and finally resumes the insertion to the correct target at $t=17 \mathrm{sec}$. The 3D Cartesian position and orientation paths for pick and insertion of both rings are plotted in Fig. 5 and 6. The orientation is visualized


Fig. 4: Ring 2 insertion: The initial target pose is wrong, leading to a collision. The robot retracts and then reattempts the insertion with the correct target.


Fig. 5: Position paths for all pick/insertions for forward motion with solid green line and reverse motion with red dotted line. During insertion at pallet 2, depicted with solid blue line, collision occurs due to having a wrong target. The robot retracts (dotted yellow line) and resumes insertion provided with the correct target.
using the quaternion $\operatorname{logarithm} \boldsymbol{\eta}=\log \left(\boldsymbol{Q} * \overline{\boldsymbol{Q}}_{0}\right)$. Both the forward and reverse motion are constrained on the same path during the course of the entire experiment.

## VI. Conclusions

In this work, a reversible Dynamic Movement Primitive (DMP) formulation was proposed which ensures global asymptotic stability of the target in both forward and backward motion. All favourable properties of the original DMP formulation are retained with a few additional desirable properties supported. The usefulness of the proposed formulation was showcased in RiH experiments.

## Appendix A - Non-Reversibility of the Original DMP AND CORRESPONDENCE WITH THE PROPOSED DMP

First we are going to examine the correspondence between the original and the proposed DMP. In particular,


Fig. 6: Orientation paths for forward and reverse motion for all pick/insertions, using $\eta=\log \left(\boldsymbol{Q} * \overline{\boldsymbol{Q}}_{0}\right)$
we will show that (1) can written in the form of (6). Let $y_{d}\left(t_{1}\right), \frac{\partial y_{d}}{\partial t_{1}}, \frac{\partial^{2} y_{d}}{\partial^{2} t_{1}}, t_{1} \in\left[t_{0, d} t_{f, d}\right]$ be the trajectory with duration $\tau_{d}=t_{f, d}-t_{0, d}$, goal $g_{d}$ and initial position $y_{d, 0}$, that is generated by the trained forcing term (5). The forcing term of (1) is thus given by:

$$
\begin{gather*}
f_{s}(x)=\frac{\tau_{d}^{2} \frac{\partial^{2} y_{d}}{\partial^{2} t_{1}}-\alpha_{z} \beta_{z}\left(g_{d}-y_{d}\right)+\alpha_{z} \tau_{d} \frac{\partial y_{d}}{\partial t_{1}}}{g_{f}(x)\left(g_{d}-y_{d, 0}\right)}  \tag{19}\\
\tau_{d} \dot{x}=h(x)  \tag{20}\\
x=x_{0, d}+H\left(t_{1} / \tau_{d}\right), t_{1} \in\left[t_{0, d} t_{f, d}\right] \tag{21}
\end{gather*}
$$

where $H(t / \tau) \triangleq \int_{t_{0} / \tau}^{t / \tau} h(x(\sigma)) d \sigma$. The dependency of $y_{d}, \frac{\partial y_{d}}{\partial t_{1}}, \frac{\partial^{2} y_{d}}{\partial^{2} t_{1}}$ on $t_{1}$ is omitted for simplicity. Substituting (19) in (1) and after simple mathematical manipulations, we get:

$$
\begin{gather*}
\ddot{y}=\ddot{y}_{r e f}-\frac{\alpha_{z}}{\tau}\left(\dot{y}-\dot{y}_{r e f}\right)-\frac{\alpha_{z} \beta_{z}}{\tau^{2}}\left(y-y_{r e f}\right)  \tag{22}\\
y_{r e f}(t) \triangleq k_{s}^{\prime}\left(y_{d}\left(t_{1}\right)-y_{d, 0}\right)+y_{0}  \tag{23}\\
\dot{y}_{r e f}(t)=k_{s}^{\prime} k_{t} \frac{\partial y_{d}\left(t_{1}\right)}{\partial t_{1}}  \tag{24}\\
\ddot{y}_{r e f}(t)=k_{s}^{\prime} k_{t}^{2} \frac{\partial^{2} y_{d}\left(t_{1}\right)}{\partial^{2} t_{1}} \tag{25}
\end{gather*}
$$

where $k_{s}^{\prime}=\left(g-y_{0}\right) /\left(g_{d}-y_{d, 0}\right)$ is the spatial and $k_{t}=$ $\tau_{d} / \tau$ the temporal scaling factor. Notice that (22)-(25) is in the form of the proposed DMP (6)-(10), since $f_{p}(x)$ is trained based on $y_{d}\left(t_{1}\right)$ hence $f_{p}(x) \approx y_{d}\left(t_{1}\right)$ and in turn $k_{s} \approx k_{s}^{\prime}$, while $\dot{f}_{p}(x) \approx \dot{y}_{d}\left(t_{1}\right)=\frac{\partial y_{d}\left(t_{1}\right)}{\partial t_{1}} \dot{t}_{1}=k_{t} \frac{\partial y_{d}}{\partial t_{1}}$, which follows from $\dot{t}_{1}=k_{t}$. In particular, solving (21) for $t_{1}$ yields $t_{1}=\tau_{d} H^{-1}(u)$, with $u=x-x_{0, d}$. Its time derivative is $\dot{t}_{1}=\tau_{d} \frac{\partial H^{-1}(u)}{\partial u} \dot{u}=\tau_{d}(h(x))^{-1} \dot{x}$ by applying the inverse function Theorem (see theorem 2.9 in [21]). Substituting $\dot{x}$ from (2) we get $\dot{t}_{1}=k_{t}$.

Remark 1: To validate that (24), (25) are the first and second order time derivatives of $y_{\text {ref }}$ notice that:

$$
\begin{equation*}
\dot{y}_{r e f}(t)=k_{s}^{\prime} \frac{d y_{d}\left(t_{1}\right)}{d t}=k_{s}^{\prime} \frac{\partial y_{d}}{\partial t_{1}} \dot{t}_{1}=k_{s}^{\prime} k_{t} \frac{\partial y_{d}}{\partial t_{1}} \tag{26}
\end{equation*}
$$

where we used the fact that $\dot{t}_{1}=k_{t}$. Accordingly, taking the time derivative of $\dot{y}_{r e f}(t)$ it follows easily that $\ddot{y}_{r e f}(t)=$ $k_{s}^{\prime} k_{t} \frac{d}{d t}\left(\frac{\partial y_{d}}{\partial t_{1}}\right)=k_{s}^{\prime} k_{t} \frac{\partial^{2} y_{d}}{\partial^{2} t_{1}} \dot{t}_{1}=k_{s}^{\prime} k_{t}^{2} \frac{\partial^{2} y_{d}}{\partial^{2} t_{1}}$.

Despite the similarity in the form between the proposed and the original DMP, their main difference is in the encoding term which in the proposed formulation utilizes only the demonstrated position trajectory while in the original, a combination of the demonstrated position velocity and acceleration trajectories are utilized (see (19)). The consequence is the non-reversibility of the original formulation as we prove in the following. Specifically, using the reverse canonical system $\tau \dot{x}=-h(x)$ of (2) and following the procedure as for deriving (22), we can conclude that (1) reduces to:

$$
\begin{equation*}
\left(\ddot{y}-\ddot{y}_{\text {rev }}\right)+\frac{\alpha_{z}}{\tau}\left(\dot{y}-\dot{y}_{\text {rev }}\right)+\frac{\alpha_{z} \beta_{z}}{\tau^{2}}\left(y-y_{r e v}\right)=-2 \frac{a_{z}}{\tau} \dot{y}_{\text {rev }} \tag{27}
\end{equation*}
$$

where $y_{\text {rev }}=k_{s}\left(y_{d}\left(\tau_{d} H^{-1}\left(x-x_{0, d}\right)\right)-y_{d, 0}\right)+y_{0}$ hence the DMP output $y, \dot{y}, \ddot{y}$ will not track faithfully the reverse trajectory due to the existence of the term $-2 \frac{a_{z}}{\tau} \dot{y}_{\text {rev }}(t)$ at the right hand-side of (27), which acts as a disturbance. It can be easily verified by simulations that the reverse DMP exhibits tracking errors and fails to reach the target at the designated time duration.

## Appendix B - Unit Quaternion Preliminaries

Given a rotation matrix $\boldsymbol{R} \in S O(3)$, an orientation can be expressed in terms of the unit quaternion $Q \in \mathbb{S}^{3}$ as $\boldsymbol{Q}=\left[w \boldsymbol{v}^{T}\right]^{T}=\left[\cos (\theta) \sin (\theta) \boldsymbol{k}^{T}\right]$, where $\boldsymbol{k} \in \mathbb{R}^{3}, 2 \theta \in$ $\left[\begin{array}{ll}0 & 2 \pi\end{array}\right)$ are the equivalent unit axis - angle representation. The quaternion product between the unit quaternions $\boldsymbol{Q}_{1}$, $\boldsymbol{Q}_{2}$ is denoted as $\boldsymbol{Q}_{1} * \boldsymbol{Q}_{2}$. The inverse of a unit quaternion is equal to its conjugate which is $\boldsymbol{Q}^{-1}=\overline{\boldsymbol{Q}}=\left[w-\boldsymbol{v}^{T}\right]^{T}$. The logarithmic $\boldsymbol{\eta}=\log (\boldsymbol{Q})$ and exponential $\boldsymbol{Q}=\exp (\boldsymbol{\eta})$ mappings $\log : \mathbb{S}^{3} \rightarrow \mathbb{R}^{3}$, exp : $\mathbb{R}^{3} \rightarrow \mathbb{S}^{3}$ respect the manifold's geometry and are defined as follows:

$$
\begin{gather*}
\log (\boldsymbol{Q}) \triangleq\left\{\begin{array}{c}
2 \cos ^{-1}(w) \frac{\boldsymbol{v}}{\|\boldsymbol{v}\|},|w| \neq 1 \\
{[0,0,0]^{T}, \text { otherwise }}
\end{array}\right.  \tag{28}\\
\exp (\boldsymbol{\eta}) \triangleq\left\{\begin{array}{c}
{\left[\cos (\|\boldsymbol{\eta} / 2\|), \sin (\|\boldsymbol{\eta} / 2\|) \frac{\boldsymbol{\eta}^{T}}{\|\boldsymbol{\eta}\|}\right]^{T},\|\boldsymbol{\eta}\| \neq 0} \\
{[1,0,0,0]^{T}, \text { otherwise }}
\end{array}\right. \tag{29}
\end{gather*}
$$

Denoting by $\boldsymbol{\Omega}=\left[\begin{array}{ll}0 & \boldsymbol{\omega}^{T}\end{array}\right]^{T}$, the time derivatives of $\boldsymbol{\eta}$, are related to $\boldsymbol{\omega}, \dot{\omega}$ as follows [16]:

$$
\begin{gather*}
\boldsymbol{\Omega}=2\left(\boldsymbol{J}_{\eta} \dot{\boldsymbol{\eta}}\right) * \overline{\boldsymbol{Q}}  \tag{30}\\
\dot{\boldsymbol{\Omega}}=2\left(\boldsymbol{J}_{\eta} \dot{\boldsymbol{\eta}}+\boldsymbol{J}_{\eta} \ddot{\boldsymbol{\eta}}\right) * \overline{\boldsymbol{Q}}-\frac{1}{2}\left[\|\boldsymbol{\omega}\|^{2} \mathbf{0}_{1 \times 3}\right]^{T}  \tag{31}\\
\boldsymbol{J}_{\eta}=\frac{1}{2}\left[\begin{array}{c}
-\sin (\theta) \boldsymbol{k}^{T} \\
\frac{\sin (\theta)}{\theta}\left(\boldsymbol{I}_{3}-\boldsymbol{k} \boldsymbol{k}^{T}\right)+\cos (\theta) \boldsymbol{k} \boldsymbol{k}^{T}
\end{array}\right]
\end{gather*}
$$

## References

[1] A. Billard, S. Calinon, R. Dillmann, and S. Schaal, Robot Programming by Demonstration. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 1371-1394.
[2] Jung-Hoon Hwang, R. C. Arkin, and Dong-Soo Kwon, "Mobile robots at your fingertip: Bezier curve on-line trajectory generation for supervisory control," in Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453), vol. 2, 2003, pp. 1444-1449 vol.2.
[3] D. Nicolis, A. M. Zanchettin, and P. Rocco, "Human intention estimation based on neural networks for enhanced collaboration with robots," in 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2018, pp. 1326-1333.
[4] S. Calinon, F. D'halluin, E. L. Sauser, D. G. Caldwell, and A. G. Billard, "Learning and reproduction of gestures by imitation," IEEE Robotics Automation Magazine, vol. 17, no. 2, pp. 44-54, 2010.
[5] D. Nguyen-Tuong, M. Seeger, J. Peters, D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, "Local gaussian process regression for real time online model learning and control," Advances in Neural Information Processing Systems 21: Proceedings of the 2008 Conference, 1193-1200 (2009), 012008.
[6] S. M. Khansari-Zadeh and A. Billard, "Imitation learning of globally stable non-linear point-to-point robot motions using nonlinear programming," in 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2010, pp. 2676-2683.
[7] P. Pastor, L. Righetti, M. Kalakrishnan, and S. Schaal, "Online movement adaptation based on previous sensor experiences," in 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems, Sept 2011, pp. 365-371.
[8] P. Pastor, H. Hoffmann, T. Asfour, and S. Schaal, "Learning and generalization of motor skills by learning from demonstration," in 2009 IEEE International Conference on Robotics and Automation, 2009, pp. 763-768.
[9] K. Mülling, J. Kober, O. Kroemer, and J. Peters, "Learning to select and generalize striking movements in robot table tennis," The International Journal of Robotics Research, vol. 32, pp. 263-279, 03 2013.
[10] J. Umlauft, D. Sieber, and S. Hirche, "Dynamic movement primitives for cooperative manipulation and synchronized motions," in 2014 IEEE International Conference on Robotics and Automation (ICRA), 2014, pp. 766-771.
[11] U. P. Schultz, J. S. Laursen, L.-P. Ellekilde, and H. B. Axelsen, "Towards a domain-specific language for reversible assembly sequences," in Reversible Computation, J. Krivine and J.-B. Stefani, Eds. Cham: Springer International Publishing, 2015, pp. 111-126.
[12] J. S. Laursen, U. P. Schultz, and L. Ellekilde, "Automatic error recovery in robot assembly operations using reverse execution," in 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2015, pp. 1785-1792.
[13] I. Iturrate, C. Sloth, A. Kramberger, H. G. Petersen, E. H. Østergaard, and T. R. Savarimuthu, "Towards reversible dynamic movement primitives," in 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2019, pp. 5063-5070.
[14] B. Nemec, M. Simonič, T. Petrič, and A. Ude, "Incremental policy refinement by recursive regression and kinesthetic guidance," in 2019 19th International Conference on Advanced Robotics (ICAR), 2019, pp. 344-349.
[15] A. J. Ijspeert, J. Nakanishi, H. Hoffmann, P. Pastor, and S. Schaal, "Dynamical movement primitives: Learning attractor models for motor behaviors," Neural Comput., vol. 25, no. 2, pp. 328-373, Feb 2013.
[16] L. Koutras and Z. Doulgeri, "A correct formulation for the orientation dynamic movement primitives for robot control in the cartesian space," in Proceedings of The 3rd Conference on Robot Learning, 30 Oct -1 Nov 2019.
[17] K. Vlachos and Z. Doulgeri, "A control scheme with a novel dmprobot coupling achieving compliance and tracking accuracy under unknown task dynamics and model uncertainties," IEEE Robotics and Automation Letters, vol. 5, no. 2, pp. 2310-2316, 2020.
[18] A. Gams, A. Ijspeert, S. Schaal, and J. Lenarcic, "On-line learning and modulation of periodic movements with nonlinear dynamical systems," Autonomous Robots, vol. 27, pp. 3-23, 072009.
[19] H. Hoffmann, P. Pastor, D. Park, and S. Schaal, "Biologically-inspired dynamical systems for movement generation: Automatic real-time goal adaptation and obstacle avoidance," in 2009 IEEE International Conference on Robotics and Automation, 2009, pp. 2587-2592.
[20] B. Nemec, N. Likar, A. Gams, and A. Ude, "Human robot cooperation with compliance adaptation along the motion trajectory," Autonomous Robots, vol. 42, 112017.
[21] H. Marquez, "Nonlinear control systems: Analysis and design," 11 2002.


[^0]:    The research leading to these results has received funding by the EU Horizon 2020 Research and Innovation Programme under grant agreement No 820767, project CoLLaboratE.
    ${ }^{1,2}$ Aristotle University of Thessaloniki, Department of Electrical and Computer Engineering, Thessaloniki 54124, Greece. antosidi@ece.auth.gr, doulgeri@eng.auth.gr

