2008 8th IEEE-RAS International Conference on Humanoid Robots
December 1 ~ 3, 2008 / Daejeon, Korea

Movement reproduction and obstacle avoidance with dynamic movement primitives and potential fields

Dae-Hyung Park, Heiko Hoffmann, Peter Pastor, and Stefan Schaal
Computer Science and Neuroscience, University of Southern California, Los Angeles, USA
daehyunp@usc.edu, heiko@clmc.usc.edu, pastorsa@usc.edu, sschaal@usc.edu

***Abstract*—Robots in a human environment need to be compliant. This compliance requires that a preplanned movement can be adapted to an obstacle that may be moving or appearing unexpectedly. Here, we present a general framework for movement generation and mid-flight adaptation to obstacles. For robust motion generation, Ijspeert et al developed the framework of dynamic movement primitives [1], [2], [3], [4], which represent a demonstrated movement with a set of differential equations. These equations allow adding a perturbing force without sacrificing stability of the desired movement. We extend this framework such that arbitrary movements in end-effector space can be represented - which was not possible before. Furthermore, we include obstacle avoidance by adding to the equations of motion a repellent force - a gradient of a potential field centered around the obstacle. In addition, this article compares different potential fields and shows how to avoid obstacle-link collisions within this framework. We demonstrate the abilities of our approach in simulations and with an anthropomorphic robot arm.**

I. Introduction

Humans can adapt a movement plan online to adjust for obstacles in the intended path. This flexibility is also required in robots operating in a human environment, where humans may move unpredictably, thus, forbidding a robot to strictly follow a preplanned path. At the same time, we like to program a robotic movement in a simple way, i.e., through demonstration.

Several researchers developed methods for learning from demonstration [5], [6]. A human movement is recorded and later reproduced by a robot. Challenges that arise for this reproduction are the correspondence problem and robustness against perturbations. The correspondence problem refers to the possible mismatch of link lengths and joints between robot and demonstrator [7]. Thus, the robot cannot directly copy the joint angles of the demonstrator. To deal with this problem, we record the human movement in task space, i.e., Cartesian space of the end-effector, and translate the resulting end-effector trajectory using inverse kinematics and dynamics models (see operational-space control).

A problem that received only minor attention is how to reproduce the movement in presence of an obstacle. We address this problem by using the dynamic movement primitive (DMP) framework, which has been successfully applied to humanoid robotics [3], [4]. A DMP can represent any recorded movement with a set of differential equations [1], [2]. Representing a movement with a differential equation has the advantage that a perturbation can be automatically corrected for by the dynamics of the system. Moreover, the DMPs are formulated in a way that convergence to a goal position is guaranteed.

For online obstacle avoidance, potential fields are a common approach [8], [9], [10]. A potential field is defined around an obstacle, and the gradient of this field results in a repellent force on the robot. This approach has been particularly popular for motion planning in mobile robotics [9], but has been also used for robotic manipulators; e.g., Brock and Khatib [11] used the potential-field method for real-time re-planning.

The dynamic movement primitive framework naturally allows the extension to potential fields. The gradient of such a field is simply added to the acceleration term of the differential equation. Thus, we can combine the representative power of DMPs with the instant computation of repellent forces.

So far, potential functions or force fields used for obstacle avoidance have been static, i.e., they compute a function of only the distance between robot and obstacle. Here, we show that using a dynamic potential field that depends on the relative velocity between end-effector and obstacle results in smoother avoidance movements.

In addition, since we work in operational space, we look at suitable strategies for computing the inverse kinematics in obstacle avoidance. Here, collisions between obstacle and manipulator links have to be avoided.

We illustrate our framework for movement reproduction and obstacle avoidance in several simulation experiments and demonstrate its feasibility and application to humanoid robotics with an anthropomorphic robot arm.

In the remainder of this article, we first explain in section II the DMP framework and show how it is modified for on-line obstacle avoidance. Section III presents the potential fields used in this article. Section IV looks at inverse-kinematics strategies suitable for obstacle avoidance with a redundant manipulator. Section V illustrates the combination of DMP and potential fields in simulation. Section VI presents experiments with the Sarcos Slave robot arm. Finally, Section VII concludes the article.

II. Dynamic movement primitives

This section briefly describes the dynamic movement primitive framework, presents a modification to the previously published work, and shows how to combine the dynamic equations with a potential field around an obstacle.

A. Original dynamic movement primitives

Dynamic movement primitives can be used to generate discrete and rhythmic movements. Here, we focus on discrete

978-1-4244-2822-9/08/$25.00 ©2008 IEEE

movements. A movement is generated by integrating the following set of differential equations[1], which can be interpreted as a linear spring system perturbed by an external force:

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

$$\tau \dot{x} = v \ , \quad (2)$$

where x and v are position and velocity for one degree-of-freedom of the system; x_0 and g are the start and goal position; τ is a temporal scaling factor; K corresponds to a spring constant; the damping term D is chosen such that the system is critically damped, and f is a non-linear function which can be adapted to allow the generation of arbitrary complex movements. This first set of equations is referred to as 'transformation system'.

The non-linear function is defined as

$$f(\theta) = \frac{\sum_i w_i \psi_i(\theta)\theta}{\sum_i \psi_i(\theta)} \ , \quad (3)$$

where ψ_i are Gaussian basis functions, $\psi_i(\theta) = \exp(-h_i(\theta - c_i)^2)$ with center c_i and width h_i, and w_i are adjustable weights. The function f does not directly depend on time; instead, it depends on a phase variable θ, which goes from 1 towards 0 during a movement and is obtained by the equation

$$\tau \dot{\theta} = -\alpha\theta \ , \quad (4)$$

where α is a predefined constant. This last differential equation is called 'canonical system'. These equations have some favorable characteristics:

- Convergence to the goal g is guaranteed since $f(\theta)$ vanishes towards the end of a movement (if the weights w_i are bounded).
- The weights w_i can be adapted to generate any desired trajectory.
- The equations are translation invariant.
- The duration of a movement can be altered simply by changing τ.

To learn a movement from demonstration, first, a movement $x(t)$ is recorded and its derivatives $v(t)$ and $\dot{v}(t)$ are computed for each time step t. Second, the canonical system is integrated, i.e., $\theta(t)$ is evaluated. Using these arrays, $f(\theta(t))$ is computed based on (1). Thus, finding the weights w_i in (3) is a linear regression problem, which can be solved efficiently.

The motion generated by a DMP is combined with a control system (Fig. 1). The above differential equations are written for a one-dimensional system; thus, for each degree of freedom (here end-effector position, orientation, and finger joint), we use one transformation system, but couple them to a single canonical system. Inverse kinematics and dynamics convert the motion generated by the transformation systems into control commands.

[1]We use a different notation than in [1], [2] to high-light the spring-like character of these equations.

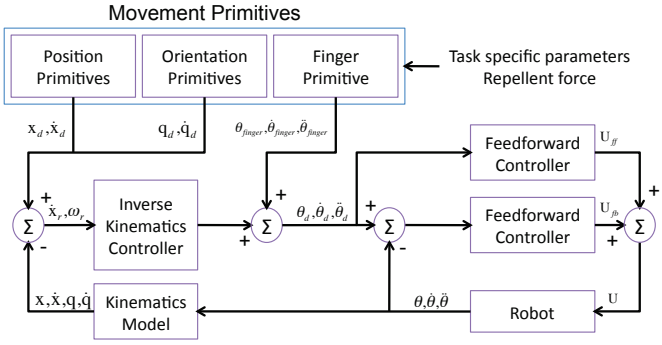

Fig. 1. Block scheme of DMP-based control. The variables $\mathbf{x}$, θ, $\mathbf{q}$, and $\mathbf{w}$ are current end-effector, joint, quaternion, and angular velocity states. $\mathbf{U}$ is a torque command. Subscript d and r mean desired and reference.

B. Modified dynamic movement primitives

We modified the DMP to overcome drawbacks in the above formulation: if start and goal position, x_0 and g, of a movement are the same, then the non-linear term in (1) cannot drive the system away from its initial state; thus, the system will remain at x_0. The scaling of f with $g - x_0$ is also problematic if $g - x_0$ is close to zero; a small change in g may lead to huge accelerations breaking the limits of the robot.

The new equations of the modified DMP are motivated from human behavioral data [12] and convergent force fields, which were observed at the frog leg after spinal-cord stimulation [13], [14]. We use three key neurophysiological findings from the frog [13]:

- After stimulating the spinal cord, a force field can be observed by measuring forces at different leg positions. These fields are often convergent.
- The magnitude of force fields is modulated in time by bell-shaped time pulses.
- Simultaneously stimulated force fields add up linearly.

These findings are realized in the model as follows. We make a first-order approximation of a convergent field around $\mathbf{w}_i$ (in 3D Cartesian space of the end-effector),

$$\boldsymbol{\chi}_i(\mathbf{x}, \mathbf{v}) = \mathbf{K}(\mathbf{w}_i - \mathbf{x}) - \mathbf{D}\mathbf{v} \ . \quad (5)$$

Each field is modulated over time with a Gaussian function centered at time c_i,

$$\psi_i(t) = \exp\left(-h(t - c_i)^2\right) \ . \quad (6)$$

We use the summation property to obtain a more complex field,

$$\mathbf{a}(\mathbf{x}, \mathbf{v}, t) = \frac{\sum_i \psi_i(t)\boldsymbol{\chi}_i(\mathbf{x}, \mathbf{v})}{\sum_i \psi_i(t)} \ . \quad (7)$$

Different from the force fields in frog, here, we use acceleration fields, i.e., $\dot{\mathbf{v}} = \mathbf{a}(\mathbf{x}, \mathbf{v}, t)$. In our robot application, we use inverse kinematics and dynamics to compute the joint torques.

Combining (5) with (7) results in the equations of motion

$$\dot{\mathbf{v}} = \mathbf{K}\left(\frac{\sum_i \psi_i(t)\mathbf{w}_i}{\sum_i \psi_i(t)} - \mathbf{x}\right) - \mathbf{D}\mathbf{v} \quad (8)$$

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

To make the equations converge to the goal $\mathbf{g}$, we add around $\mathbf{g}$ another field (5) and shift the weight from (8) to the new field. As weight, we use the phase variable θ, as computed by (4), and thus, the new acceleration field becomes

$$\dot{\mathbf{v}} = \theta\,\mathbf{K}\left(\frac{\sum_i \psi_i(\theta)\mathbf{w}_i}{\sum_i \psi_i(\theta)} + \mathbf{x}_0 - \mathbf{x}\right) + (1-\theta)\mathbf{K}(\mathbf{g}-\mathbf{x}) - \mathbf{D}\mathbf{v} \ . \quad (10)$$

We inserted an extra $\mathbf{x}_0$ to make the equation translation invariant. Furthermore, we changed the dependence of ψ on t to θ. As in the original DMP, this change will allow more flexibility since we can manipulate θ; absolute timing cannot be easily modified.

We rewrite (10) to make it look similar to (1) and include τ, as before, to make the equation invariant with respect to movement duration. Thus, the transformation system becomes

$$\tau\dot{\mathbf{v}} = \mathbf{K}(\mathbf{g}-\mathbf{x}) - \mathbf{D}\mathbf{v} - \mathbf{K}(\mathbf{g}-\mathbf{x}_0)\theta + \mathbf{K}\mathbf{f}(\theta) \quad (11)$$

$$\tau\dot{\mathbf{x}} = \mathbf{v} \ . \quad (12)$$

Again, we use the same canonical system as before. Towards the end of a movement, θ approaches 0, as before. Thus, convergence to the goal $\mathbf{g}$ is again guaranteed.

This new formulation can learn movements with the same start and goal position. Furthermore, the non-linear term that contains $\mathbf{f}$ does not scale anymore with $\mathbf{g}-\mathbf{x}_0$; thus, small values of $\mathbf{g}-\mathbf{x}_0$ do not cause huge accelerations anymore when changing $\mathbf{g}$ (Fig. 2). We use these equations to combine dynamic movement primitives with potential fields for obstacle avoidance.

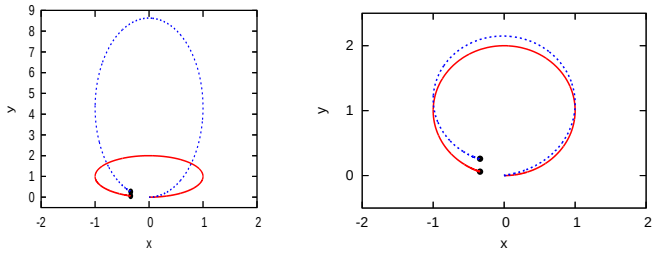

Fig. 2. Comparison of goal adaptation between original (Left) and new (Right) DMP formulation for a 2-dimensional movement in (x, y). The same demonstrated movement (solid curve) and goals (black dots) are used for both formulations. The dashed curves show the results of changing the goal $\mathbf{g}$ for the entire movement (without changing w_{ik}).

C. Combining a modified DMP with a potential field

For combining a DMP with a potential field for obstacle avoidance, we generate movements in operational space. Thus, the variable $\mathbf{x}$ describes the end-effector position. An obstacle's position and potential field are more easily available in end-effector rather than joint space. Thus, using a DMP in operational space allows a simple addition of a repellent acceleration term $\boldsymbol{\varphi}(\mathbf{x}, \mathbf{v})$ to the transformation system (11),

$$\tau\dot{\mathbf{v}} = \mathbf{K}(\mathbf{g}-\mathbf{x}) - \mathbf{D}\mathbf{v} - \mathbf{K}(\mathbf{g}-\mathbf{x}_0)\theta + \mathbf{K}\mathbf{f}(\theta) + \boldsymbol{\varphi}(\mathbf{x}, \mathbf{v}) \quad (13)$$

$$\tau\dot{\mathbf{x}} = \mathbf{v} \ . \quad (14)$$

The repulsive acceleration is the negative gradient of a potential field, which depends on the relative position and velocity of the end-effector to the obstacle. This addition allows the generated path of a movement to be changed by the properties of the potential function.

III. Potential Fields for Obstacle Avoidance

Starting from a static potential field, we develop a new potential field, a dynamic field that takes into account the relative velocity between end-effector and obstacle. In this article, we consider only a single point obstacle.

A. Static potential field

The potential-field method was proposed by Khatib [10] and Krogh [8]. By design, each obstacle creates a potential field $U(\mathbf{x})$ at the position $\mathbf{x}$ of a point moving in the field. In our case, the additional term in (13) is given by the gradient of the field, i.e., $\boldsymbol{\varphi}(\mathbf{x}) = -\nabla U(\mathbf{x})$. Khatib's potential field is static since it depends only on the distance $p(\mathbf{x})$ between current position and obstacle,

$$U_{\text{static}}(\mathbf{x}) = \left\{ \begin{array}{ll} \frac{\eta}{2}\left(\frac{1}{p(\mathbf{x})} - \frac{1}{p_0}\right)^2 & : \ p(\mathbf{x}) \le p_0 \\ 0 & : \ p(\mathbf{x}) > p_0 \end{array} \right\} \ , \quad (15)$$

where p_0 is the radius of influence of the obstacle, and η is a constant gain.

We found that this potential field did not allow smooth obstacle avoidance since it ignores the end-effector's speed and direction of movement. Thus, we propose a new potential field.

B. Dynamic potential field

The dynamic potential field is a function of the end-effector position $\mathbf{x}$ and its velocity $\mathbf{v}$. Because of this velocity dependence, we call this field 'dynamic'. We design the dynamic potential field to achieve the following properties:

1) The magnitude of the potential decreases with the distance from $\mathbf{x}$ to the obstacle.
2) The magnitude of the potential increases with the speed of $\mathbf{x}$ and is zero when the speed of $\mathbf{x}$ is zero.
3) The magnitude of potential decreases with the angle between the current velocity direction of $\mathbf{x}$ and the direction towards the obstacle. In addition, the magnitude is zero, if the angle is over $90°$ (i.e., end-effector moves away from the obstacle).

Based on these arguments, we define the dynamic potential function as

$$U_{\text{dyn}}(\mathbf{x}, \mathbf{v}) = \left\{ \begin{array}{ll} \lambda(-cos\theta)^{\beta}\frac{||\mathbf{v}||}{p(\mathbf{x})} & : \ \frac{\pi}{2} < \theta \le \pi \\ 0 & : \ 0 \le \theta \le \frac{\pi}{2} \end{array} \right\} \quad (16)$$

where λ is a constant for the strength of the entire field, β is a constant. For all experiments in this article, we chose $\beta = 2$. The angle θ is taken between the current velocity $\mathbf{v}$ and the end-effector position $\mathbf{x}$ relative to the position of the obstacle,

$$\cos\theta = \frac{\mathbf{v}^T\mathbf{x}}{||\mathbf{v}||\,p(\mathbf{x})} \quad . \tag{17}$$

The angle θ is limited to the range from 0 to π. Figure 3 shows the potential field in 2D end-effector space assuming equal velocity for each position $\mathbf{x}$. For moving obstacles, we use as input $(\mathbf{x}, \mathbf{v})$ to the potential function (16) the position and velocity *relative* to the obstacle.

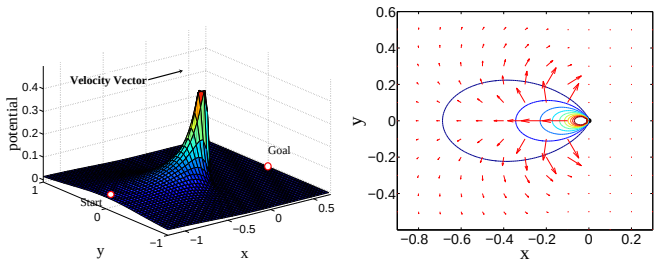

Fig. 3. Dynamic potential field in two-dimensional task space.

The obstacle force is derived from a negative gradient of the potential function as

$$\begin{aligned}\varphi(\mathbf{x},\mathbf{v}) &= -\nabla_x U_{\text{dyn}}(\mathbf{x},\mathbf{v}) \\ &= \lambda(-\cos\theta)^{\beta-1}\frac{||\mathbf{v}||}{p}\left(\beta\,\nabla_x\cos\theta - \frac{\cos\theta}{p}\nabla_x p\right)\end{aligned} \tag{18}$$

where $\pi/2 < \theta \leq \pi$. The dynamic movement primitive can be modified with this dynamic potential field.

IV. Avoiding Link Collisions

The end-effector is guided by the artificial potential field around an obstacle. However, planning in end-effector space may not prevent collisions between robot links and obstacles. To solve such a problem, various link-collision avoidance mechanisms were proposed. Khatib proposed to put an adequate number of points, which are subjected to the potential field, on the manipulator links [10]. The resulting sum of forces shields entire links from obstacles. Maciejewski and Klein proposed an inverse-kinematics model in which the null-space is constrained to avoid link collisions [15]. We combined the Maciejewski and Klein approach with our framework.

In the following, we briefly describe Maciejewski's method. The general solution for velocity based inverse kinematics is given by

$$\dot{\boldsymbol{\theta}} = \mathbf{J}^+\dot{\mathbf{x}} + (\mathbf{I} - \mathbf{J}^+\mathbf{J})\boldsymbol{\xi} \quad , \tag{19}$$

where $\dot{\boldsymbol{\theta}}$ is the joint velocity vector, $\dot{\mathbf{x}}$ is the end-effector velocity vector (here, as provided by the DMP), and $\mathbf{J}$ and $\mathbf{J}^+$ are the Jacobian matrix and its pseudo inverse. The first term on the right side of the equation shows the least squares solution. The second term is the null-space movement with arbitrary velocity $\boldsymbol{\xi}$. This null-space movement is adapted for obstacle avoidance.

Let $\mathbf{x}_o$ be the closest point to the obstacle (Fig. 4) and $\mathbf{J}_o$ the corresponding Jacobian, then

$$\dot{\mathbf{x}}_o = \mathbf{J}_o\,\dot{\boldsymbol{\theta}} \quad . \tag{20}$$

We choose for the null-space constraint, a movement $\dot{\mathbf{x}}_o$ away from the obstacle, here, $\dot{\mathbf{x}}_o = -\text{const} \cdot \nabla U_{\text{static}}(\mathbf{x}_o)$. Using (19), we substitute $\dot{\boldsymbol{\theta}}$ in (20) and obtain the desired null-space velocity $\boldsymbol{\xi}$,

$$\boldsymbol{\xi} = \left[\mathbf{J}_o(\mathbf{I} - \mathbf{J}^+\mathbf{J})\right]^+\left(\dot{\mathbf{x}}_o - \mathbf{J}_o\,\mathbf{J}^+\dot{\mathbf{x}}\right) \quad . \tag{21}$$

This result is substituted back into (19) and simplified [15]:

$$\dot{\boldsymbol{\theta}} = \mathbf{J}^+\dot{\mathbf{x}} + \left[\mathbf{J}_o(\mathbf{I} - \mathbf{J}^+\mathbf{J})\right]^+\left(\dot{\mathbf{x}}_o - \mathbf{J}_o\,\mathbf{J}^+\dot{\mathbf{x}}\right) \tag{22}$$

This method can be directly applied to our approach of combining dynamic movement primitives with potential fields. Since we plan in operational space, we can use Maciejewski's method directly for the inverse kinematics.

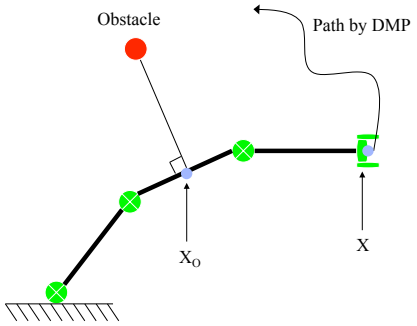

Fig. 4. Location of $\mathbf{x}_o$ and $\mathbf{x}$ on the manipulator.

The question arises which point to put into the potential field equation (16), the end-effector end-point $\mathbf{x}$ or the closest point on the manipulator to the obstacle, $\mathbf{x}_o$. We experimented with both alternatives. In the case of $\mathbf{x}$, the end-effector space planning is unaffected by the link avoidance algorithm. In the case of $\mathbf{x}_o$, the DMP has also feedback from the link states. Compared with Khatib's approach, the potential field does not directly affect the joint states, but it makes the DMP change the end-effector movement, which helps the closest link to detour the obstacle.

V. Simulation

We first use two-dimensional point movements to demonstrate the effect of a potential field on the dynamic movement primitives. Second, we demonstrate avoiding a moving obstacle in end-effector space and with a 7-degree-of-freedom robot arm simulator. Third, we used this simulator and a three-link planar arm to demonstrate the avoidance of link collisions.

A. General methods

As desired trajectories, we generated minimum-jerk movements between a start and goal point. We used these trajectories as 'demonstrated movements', from which, the weights w_i of a DMP are obtained as described in Section II-A. The DMPs could reproduce these movements accurately. Finally, a movement was generated under the influence of a potential field.

B. Moving point with fixed obstacle

First, we consider only moving points, no links. Figure 5 and 6 show the simulation result for a collision-free movement of an 'end-effector' imitating a learned movement. In this simulation, a fixed point obstacle is located on the center of the work space. The target end-effector movement passes through the obstacle. The dynamic potential field produces smoother movements than the static potential field.

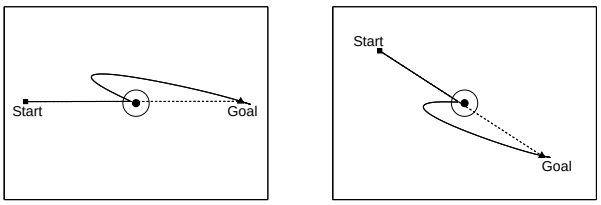

Fig. 5. Movement generation with static potential field (solid curve). The obstacle is in the center of each graph. The target movement is shown as dashed line.

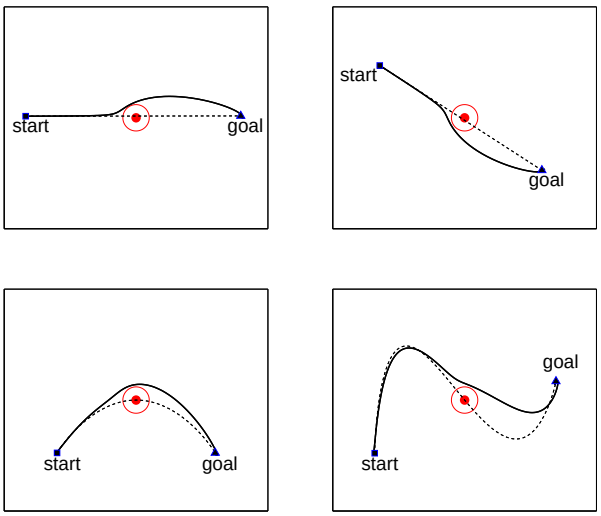

Fig. 6. Movement generation with dynamic potential field (solid curve). The obstacle is in the center of each graph. The target movement is shown as dashed curve.

C. Avoiding a moving obstacle

Our obstacle avoidance approach allows online adaptation to moving obstacles. Figure 7 shows the simulation result for a moving obstacle. Initially, a point obstacle is located on the left bottom of the work space, and then it moves toward the right top with constant velocity. The conventional static potential method shows an unstable avoidance movement, which oscillates. On the other hand, the dynamic potential method results in a smooth obstacle-avoidance movement.

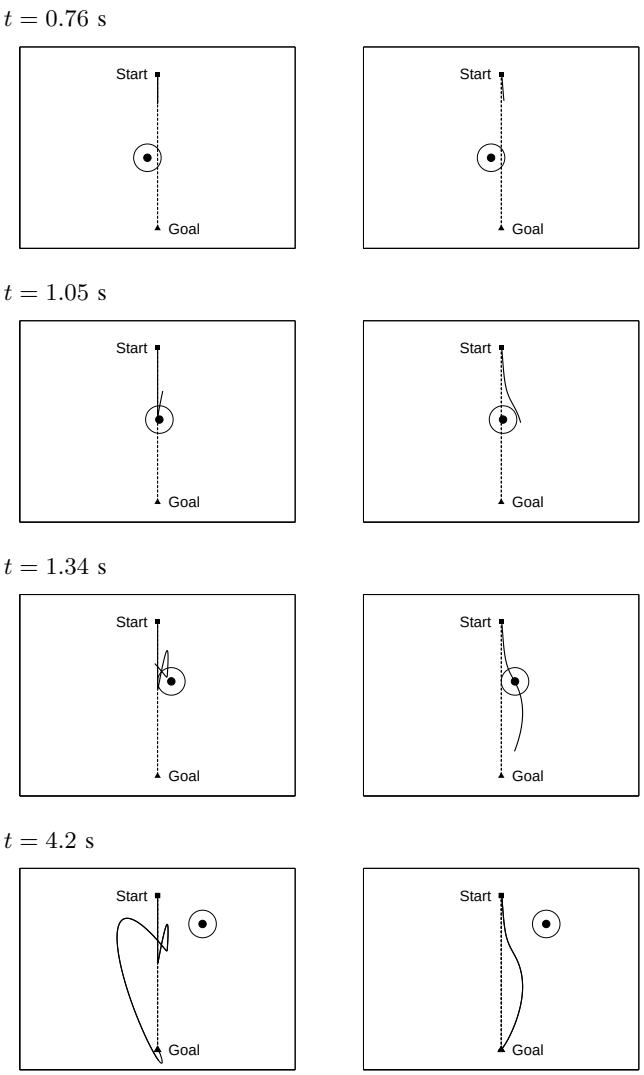

Fig. 7. Movement generation with moving obstacle comparing static (Left column) with dynamic potential field (Right column). The target movement is shown with a dashed line.

In the following, we use only the dynamic potential field. Figure 8 shows avoiding a moving obstacle with a simulated robot arm. This arm has the dimensions and degrees of freedom of our Sarcos Slave robot arm used in the experiment in Section VI. The simulated robot arm could avoid the moving obstacle.

D. Link-collision avoidance

We study the link-collision avoidance first with a three-link-planar-arm simulation and second with a simulated 7 degree-of-freedom manipulator in 3D space (see Fig. 8). Figure 9 shows the three-link-manipulator movements with DMP and

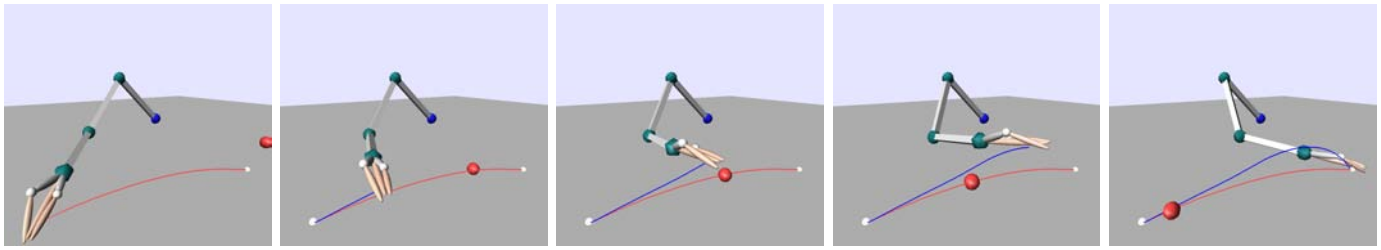

Fig. 8. Seven-degree-of-freedom robot arm avoiding a moving obstacle (red ball) while trying to reproduce a demonstrated end-effector trajectory. Demonstrated trajectory (red curve) and reproduction with dynamic potential field (blue curve) are shown.

dynamic potential field. Without a link-avoidance algorithm, a collision occurs between link and obstacle (Figure 9 A). This collision can be avoided by using the algorithm described in Section IV (Figures 9 B and C). For this algorithm, we tested two variants. If using the end-effector $\mathbf{x}$ as input into the potential field, the end-effector moved closer to the target line than if using $\mathbf{x}_o$ as input. In the case of $\mathbf{x}_o$, the movement is detoured by the link interaction.

We use the simulated 7 degree-of-freedom manipulator for a quantitative test to compare between the two alternatives for link-avoidance (Section IV). In the test, the end-effector is moved along a predefined curve, and the obstacle is put at random positions in between robot base and end-effector trajectory. Depending on the strength, λ, of the potential field, we can make either the error of the end-effector small relative to the unperturbed movement - the error is computed as normalized mean squared error (nMSE) between two curves - or we increase the minimum distance between manipulator and obstacle. Thus, we have a trade-off between these two optimization criteria. We plotted both of these criteria in one diagram for various values of λ (Fig. 10). For each λ value and condition (input $\mathbf{x}$ or $\mathbf{x}_o$), we simulated 100 movements. In the graph (Fig. 10), we used only the data of the successful runs, in which the end-effector reached the goal position within a given time window and without collision.

In Fig, 10, we compare the alternatives $\mathbf{x}$ and $\mathbf{x}_o$ as input for the potential field. For $\mathbf{x}$, we got smaller nMSE values for given minimum distances to the obstacle. Thus, as above, using $\mathbf{x}$ as input gave the better result. However, if we compare the robustness of the obstacle-avoidance movement, we got a different result. Here, robustness means the percentage of successful trials. This percentage was 96.7% for $\mathbf{x}_o$ and 75.9% for $\mathbf{x}$ as input.

VI. Robot Experiment

We applied our framework for object manipulation with a redundant robot arm.

A. Methods

Our experimental platform is a seven-DOF hydraulically actuated robot arm (Sarcos Slave Arm, from Sarcos, Salt Lake City). Its range is comparable to that of a human arm. As obstacle, we use one small red ball of 2.3 cm radius; its coordinates were obtained beforehand. In future work, we plan to use a visual system to extract the coordinates of this ball.

To achieve object manipulation, we combined three components: position, orientation, and finger primitives for the end-effector (Fig. 1). Position primitives describe the end-effector movement in Cartesian space; orientation primitives describe the gripper orientation in quaternion space, and a finger primitive is used for the finger joint angle. Thus, we used eight transformation systems, and each transformation system provides desired states, such as position, velocity, and acceleration. All transformation systems are linked to one canonical system (Section II-A). The desired position, quaternion, and angle states are mapped onto control commands using inverse kinematics and dynamics (Fig. 1).

For position and orientation control, we used a velocity based inverse kinematics controller with position and quaternion feedback, which uses a unit quaternion $\mathbf{q}$,

$$\mathbf{q} = \begin{bmatrix} \eta \\ \epsilon \end{bmatrix} = \begin{bmatrix} \eta & \epsilon_1 & \epsilon_2 & \epsilon_3 \end{bmatrix}^T \ , \tag{23}$$

where η and ϵ are the scalar and vector parts. An end-effector orientation can be described by a direction vector $\mathbf{r}$ (with $\|\mathbf{r}\| = 1$) and a rotation, φ, around this vector. The corresponding quaternion is given by

$$\eta = \cos\left(\frac{\varphi}{2}\right) \ , \tag{24}$$

$$\epsilon = \mathbf{r}\sin\left(\frac{\varphi}{2}\right) \ . \tag{25}$$

To use an inverse kinematics algorithm based on a unit quaternion, a suitable orientation error is formulated as

$$\mathbf{e}_0 = \eta_d\epsilon - \eta\epsilon_d - \epsilon_d \times \epsilon \ , \tag{26}$$

where (η_d, ϵ_d) and (η, ϵ) are the desired and current orientation represented as quaternions.

Therefore, the task space reference velocity $\dot{x}_r$ and angular velocity w_r in Fig. 1 can be computed as

$$\dot{x}_r = \dot{x}_d + \mathbf{K}_p(x_d - x) \tag{27}$$

$$\begin{aligned} \omega_r &= \omega_d - \mathbf{K}_o\mathbf{e}_o \\ &= \omega_d - \mathbf{K}_o\left(\eta_d\epsilon - \eta\epsilon_d - \epsilon_d \times \epsilon\right), \end{aligned} \tag{28}$$

where w_d is desired angular velocity. $\mathbf{K}_p$ and $\mathbf{K}_o$ are feedback gain constants.

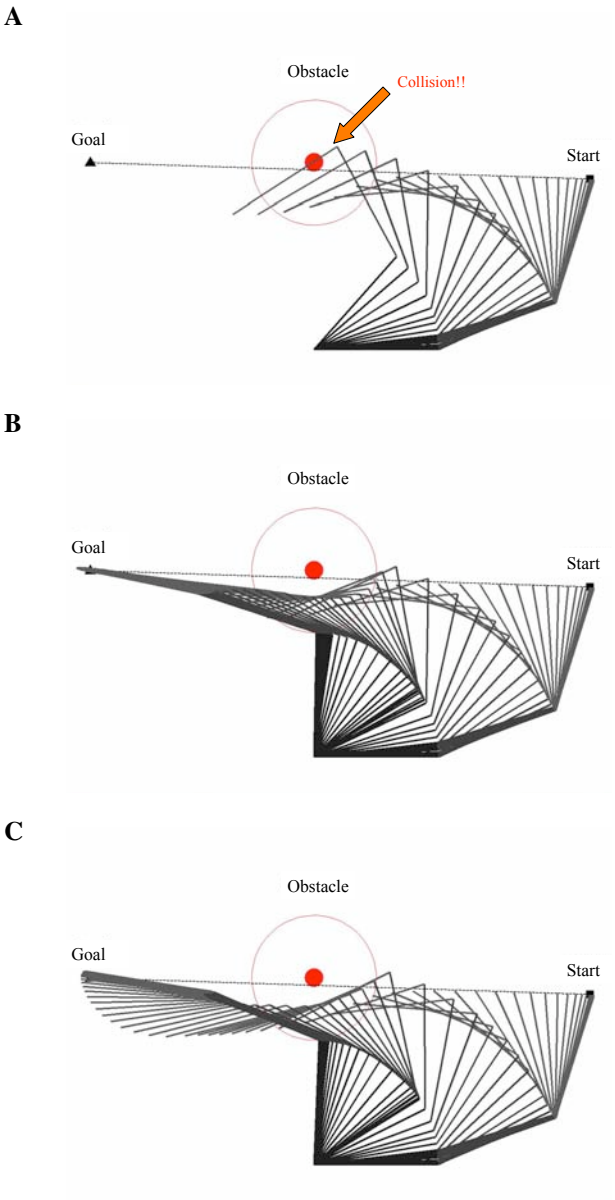

Fig. 9. Three-link manipulator avoiding a point obstacle. (A) Without avoiding link collisions. (B) and (C) using Maciejewski and Klein's algorithm [15] to avoid link collisions, in (B), using the end-effector position $\mathbf{x}$ as input into the dynamic potential field and in (C), using the closest point $\mathbf{x}_o$ on the manipulator as input.

To imitate human manipulation, a teacher first demonstrated the desired movement using the Sarcos Master robot arm, and we recorded the required parameters, such as, position, orientation, and finger angle. For the robot experiment in this article, we recorded a grasping, a placing, and a releasing movement. Then, for each DMP, we obtained the weights w_i (Section II-A) such that the DMP reproduced the demonstrated motion. For obstacle avoidance, as before, we added the repellent force from our dynamic potential field to the DMP.

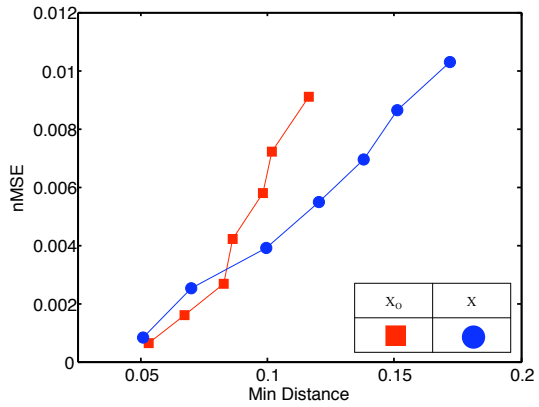

Fig. 10. Trade off between minimum distance to obstacle and nMSE along tracked trajectory for various values of λ, the strength of the dynamic potential field. The two alternatives $\mathbf{x}$ and $\mathbf{x}_o$ as input to the potential field are compared.

B. Results

The robot arm could grasp an object, place it to a different location, and release it again. During the placing motion, the robot smoothly avoided an obstacle in mid-flight using the dynamic potential field (Fig. 11 and 12).

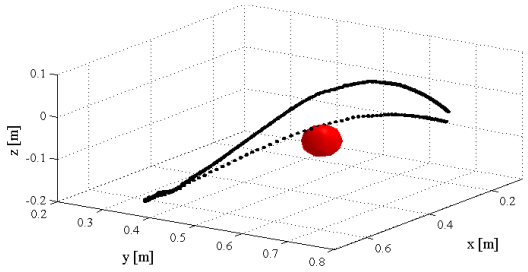

Fig. 11. End-point trajectory for the robot experiment in Fig. 12. The dashed curve shows the movement without obstacle and the solid curve with obstacle.

VII. Conclusions and Future Work

We combined the dynamics movement primitive (DMP) framework with potential fields for online obstacle avoidance. DMP provided us with a framework to reproduce a movement from human demonstration. Since the movement is represented with a set of differential equations, it has attractor dynamics, and a perturbation can be automatically corrected for. As perturbation, for obstacle avoidance, we added a repulsive force around the obstacle. Thus, the robot could avoid the obstacle while being attracted to the original demonstrated movement. This article is the first that demonstrates this combination and shows its successful application in a real robot.

Furthermore, we introduced a new potential field that together with DMPs results in smooth obstacle avoidance. Interestingly, in comparison, a static potential field could not produce smooth movements, particularly, if the end-effector directly approached the obstacle. These movements seem to

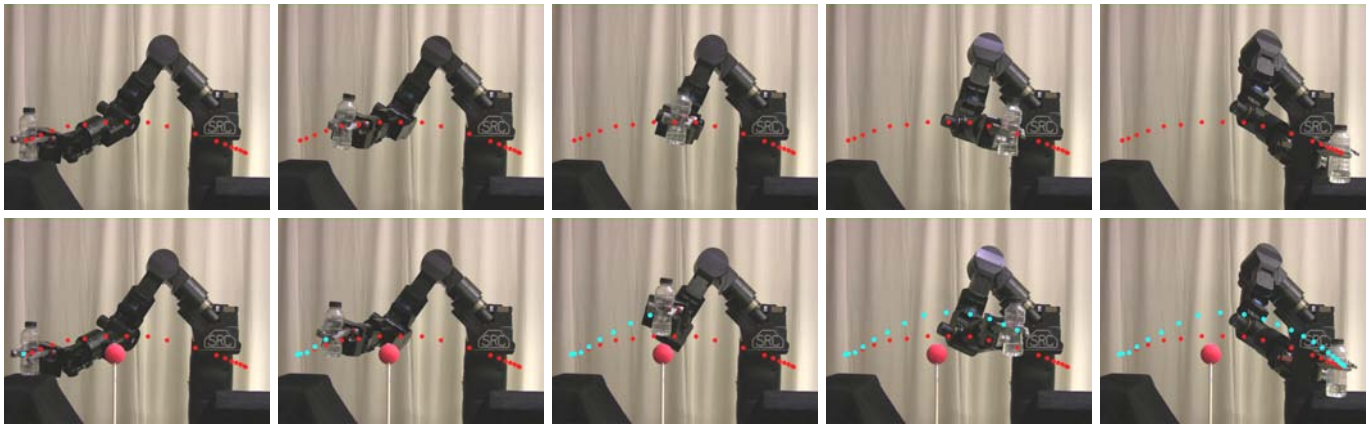

Fig. 12. Obstacle avoidance with the Sarcos Slave Arm. (Top row) Reproduction of a demonstrated movement; (Bottom row) result of obstacle avoidance with potential field. Red dots show the reproduced end-effector trajectory from the DMP, and blue dots show the modified end-effector trajectory by the potential field.

require a dependence on the relative velocity between obstacle and end-effector.

This article studied only a single point obstacle. Complex obstacles may be described using several points. For many obstacles, however, the superposition of potential fields may contain local minima. Alternatively, a potential field could be chosen to match the shape of the obstacle. A solution to this problem is part of future research.

We represented a movement in operational space. The DMP described the movement of the end-effector. Thus, we needed inverse kinematics to map a movement onto the joint angles. A null-space optimization was used that avoids collisions of the manipulator links with the obstacle. The potential field takes as one of its inputs the position of the end-effector. As alternative input to the potential field, since link-avoidance requires the computation of the closest point on the manipulator to the obstacle, we tested using this closest point instead of the end-effector. The advantage of using the closest point was robustness; more movements reached the goal without collision and within a given time. On the other hand, tracking a demonstrated movement was better when using the end-effector as input to the potential field.

We plan to use the presented framework for movement generation and obstacle avoidance in a full-body humanoid robot. Thus, besides the arms, also head and legs could use dynamic potential fields for obstacle avoidance. Furthermore, we plan to use visual feed-back to detect an obstacle's position. With this feed-back, our framework allows real-time reaction to moving obstacles.

ACKNOWLEDGMENT

This research was supported by NSF, DFG, and NASA.

REFERENCES

[1] A. J. Ijspeert, J. Nakanishi, and S. Schaal, "Movement imitation with nonlinear dynamical systems in humanoid robots," in *International Conference on Robotics and Automation*. Washington, DC: IEEE, 2002, pp. 1398–1403.

[2] ——, "Learning attractor landscapes for learning motor primitives." in *Advances in Neural Information Processing Systems*, S. Becker, S. Thrun, and K. Obermayer, Eds., vol. 15. MIT Press, Cambridge, MA, 2003, pp. 1523–1530.

[3] S. Schaal, J. Peters, J. Nakanishi, and A. Ijspeert, "Control, planning, learning, and imitation with dynamic movement primitives," in *Workshop on Bilateral Paradigms on Humans and Humanoids, IEEE International Conference on Intelligent Robots and Systems*, Las Vegas, NV, 2003.

[4] S. Schaal, "Dynamic movement primitives: A framework for motor control in humans and humanoid robotics," in *2nd International Symposium on Adaptive Motion of Animals and Machines (AMAM)*, 2003.

[5] C. G. Atkeson and S. Schaal, "Robot learning from demonstration," in *International Conference on Machine Learning*, 1997.

[6] A. Billard and R. Siegwart, "Robot learning from demonstration," *Robotics and Autonomous Systems*, vol. 47, pp. 65–67, 2004.

[7] C. L. Nehaniv and K. Dautenhahn, "The correspondence problem," in *Imitation in animals and artifacts*, K. Dautenhahn and C. L. Nehaniv, Eds. Cambridge, MA, USA: MIT Press, 2002, pp. 41–61.

[8] B. H. Krogh, "A generalized potential field approach to obstacle avoidance control," in *International Robotics Research Conference*, Bethlehem, PA, 1984.

[9] J. Borenstein and Y. Koren, "Real-time obstacle avoidance for fast mobile robots," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 19, pp. 1179–1187, 1989.

[10] O. Khatib, "Real-time obstacle avoidance for manipulators and fast mobile robots," *International Journal of Robotics Research*, vol. 5, p. 90, 1986.

[11] O. Brock and O. Khatib, "Real-time re-planning in high-dimensional configuration spaces using sets of homotopic paths," in *Proceedings of the International Conference on Robotics and Automation*, vol. 1. IEEE, 2000, pp. 550–555.

[12] H. Hoffmann and S. Schaal, "Human movement generation based on convergent flow fields: a computational model and a behavioral experiment," in *Advances in Computational Motor Control VI*, R. Shadmehr and E. Todorov, Eds., San Diego, CA, 2007.

[13] S. F. Giszter, F. A. Mussa-Ivaldi, and E. Bizzi, "Convergent force fields organized in the frog's spinal cord," *Journal of Neuroscience*, vol. 13, no. 2, pp. 467–491, 1993.

[14] H. Hoffmann, P. Pastor, and S. Schaal, "Dynamic movement primitives for movement generation motivated by convergent force fields in frog," in *Fourth International Symposium on Adaptive Motion of Animals and Machines*, R. Ritzmann and R. Quinn, Eds., Case Western Reserve University, Cleveland, OH, 2008.

[15] A. A. Maciejewski and C. A. Klein, "Obstacle avoidance for kinematically redundant manipulators in dynamically varying environments," *International Journal of Robotics Research*, vol. 4, pp. 109–116, 1985.