



Learning Coupling Terms for Obstacle Avoidance

Submitted by
Akshara Rai

Supervisors
Franziska Meier
Auke Ijspeert
Stefan Schaal



**Computational Learning &
Motor Control Lab**

BIOROB
EPFL Biorobotics Laboratory

Master Thesis, Academic year 2013-2014, Spring Semester

ABSTRACT

Autonomous manipulation in dynamic environments is important for robots to perform everyday tasks. For this, a manipulator should be capable of interpreting the environment and accordingly planning an appropriate movement. At least two possible approaches exist for this in literature. Most commonly, a planning system is used to generate a complex movement. This plan takes all the constraints in the environment into account and finds the globally optimal solution. Alternatively, a simple plan could be chosen and modified with sensory feedback to accommodate additional constraints. One way of doing this is to equip the controller with features that remain dormant most of the time, except when specific situations arise, e.g., invoking obstacle avoidance behaviour on detecting an obstacle in the surrounding. Dynamic Movement Primitives (DMPs) form a robust and versatile starting point for such a controller that can be modified online using a non-linear term, called the coupling term. This can prove to be a fast and reactive way of obstacle avoidance in a human-like fashion. We propose a method to learn this coupling term from human demonstrations using simple features as a starting point and making it more robust to avoid a larger range of obstacles. We also test the ability of our coupling term to model different kinds of obstacle avoidance behaviours in humans and use this learnt coupling term to avoid obstacles in a reactive manner. This line of research aims at pushing the boundary of reactive control strategies to more complex scenarios, such that complex and usually computationally more expensive planning methods can be avoided as much as possible.

ACKNOWLEDGEMENTS

First and foremost, I would like to express profound gratitude to Franziska Meier, without whom this work couldn't have been possible. Thank you Franzi, for not just supervising me, but also making sure that I clearly understood what I was doing and did it correctly. Her support, guidance, advice as well as pain-staking efforts at debugging my codes, and reading my drafts were far beyond what I had expected. I cannot imagine a better mentor, supervisor and friend for a Master student who decides to go to a completely new environment to pursue his/her work.

I would also like to thank Prof. Stefan Schaal, who gave me the independence to do what I chose, while also making sure that I did not get lost in the vast expanse of ideas. My discussions with Stefan, be it about work, or over champagne, were among some of the best parts of my thesis, where I got a small peek into his brilliant mind. I hope that I continue to receive his guidance and advice in the future.

I thank my fellow labmates, at USC and Tuebingen, for never making me feel like an outsider, even when I was one. Thanks Nick, for all the data I collected from you, and much more. Thanks Sean, Manuel, Daniel, Alex, Meghan, Bharath, Ludo, Jeanette, Felix, and basically everyone at both the labs for making this such a wonderful experience. I am really sad to leave the lab, but its a small world. We will stay connected, thanks to the big net.

Last but not the least, I thank my friends Rupesh and Arzoo for making the good times better and the not-so-good ones bearable. Thank you for being there when I needed you.

CONTENTS

| | | |
|----------|--|-----------|
| 1 | Introduction | 9 |
| 2 | Related work on obstacle avoidance | 13 |
| 3 | Background on Dynamic Movement Primitives | 15 |
| 3.1 | Modelling and generating movement with DMPs | 16 |
| 3.2 | Coupling terms | 17 |
| 4 | Coupling Terms for Obstacle Avoidance | 19 |
| 5 | Learning Coupling Term | 23 |
| 5.1 | Basis of features | 24 |
| 5.1.1 | ϕ_1 and ϕ_2 | 24 |
| 5.1.2 | ϕ_3 | 25 |
| 5.2 | Linear Regression to learn the coupling terms | 25 |
| 5.3 | Automatic Relevance Determination | 26 |
| 6 | Convergence to goal | 29 |
| 7 | Experiments | 31 |
| 7.1 | Evaluation of the coupling term on simulated examples | 31 |
| 7.2 | Coupling term from human demonstrations | 32 |
| 7.3 | Modelling human behaviour | 35 |
| 7.3.1 | Limitations in modelling different behaviours | 36 |
| 7.4 | Unrolling coupling term from human demonstrations | 37 |
| 7.5 | Automatic Relevance Determination on basis of features | 40 |
| 7.6 | Experiments on a simulated robot arm | 41 |

| | | |
|----------|--|-----------|
| 8 | Future work | 43 |
| 9 | Conclusion | 45 |
| A | Appendix | 47 |
| A.1 | Experiments on a simulated robot arm | 47 |
| | Bibliography | 53 |

CHAPTER 1

INTRODUCTION

To become part of daily human life, humanoid robots need to deal with dynamic and stochastic environments that occur in everyday settings. These environments are by no means fixed, or even completely known and are constantly undergoing changes. Robots should be able to respond as well as adapt to such changes. In response to a change in the environment, the robot can pursue at least two different strategies. First, it could re-plan, while anticipating future changes in the environment with model-based predictions. Such planning usually takes into account the whole set-up of the environment and searches for a globally optimal solution. It is often computationally expensive and time consuming, such that rapid reactions are hard to accomplish. Alternatively, the robot could try to modify its on-going control policy with reactive strategies, which are usually very fast to compute, but sub-optimal in comparison to a global planning algorithm. In this thesis, we will pursue the latter strategy and examine ways of learning one such strategy for obstacle avoidance from demonstrations.

Dynamic Movement Primitives (DMPs), as introduced in [6] provide a simple and versatile framework to approach this problem. DMPs are differential equations with well-defined attractor properties that can model almost arbitrarily complex motions. They can be easily initialized with imitation learning and used to generate movements that are inherently robust to changes in task duration, goal, starting point, translation and slight perturbations. These features make DMPs an extremely powerful tool for performing tasks in dynamic environments. Our work exploits the ability of DMPs to be modified online based on feedback from the environment using non-linear functions called coupling terms.

In order to address a reactive controller for dynamic environments using DMPs,

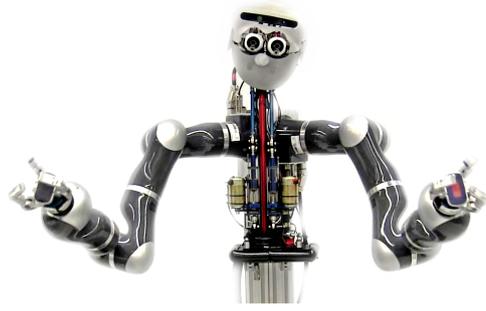


Figure 1.1: The robot *Apollo* is a dual arm manipulation platform consisting of two 7 DoF KUKA light weight arms, two Barrett hands and a SARCOS humanoid head.

we need to generate sufficiently powerful coupling terms that can use relevant information from the environment and change a motion plan accordingly. For the nominal movement of the DMP, i.e., without unexpected events in the environment, the coupling term should be inactive, i.e., the features that form the coupling term should be dormant. Only if certain sensory events trigger the features to be non-zero, the coupling term will become active and modulate the on-going movement appropriately. With the right features, very powerful feedback terms can be added to the DMP controller. Therefore, designing appropriate features, either manually, or hopefully with general machine learning techniques, is at the core of this approach. We envision that daily task components like obstacle avoidance, avoidance of joint-angle limits, force control when contacting an object, bi-manual task coordination, etc., can become part of reactive feedback control.

As a first step towards this vision of reactive feedback controllers, we will examine obstacle avoidance. To avoid obstacles, a manipulator needs to modify an original plan of movement based on position, speed and size of the obstacle. It might often not know the exact location, or the complete shape of the obstacle. We develop a reactive, local method of obstacle avoidance using coupling terms that modifies an initially planned DMP online. We start by using an initially simple obstacle avoidance formulation, as proposed in [21] for 2-D point obstacle avoidance by humans, and transform it into a 3-D formulation for extended and multiple obstacles. We go on to propose a basis of features that can be used to avoid obstacles of arbitrary sizes. Then using these features, we learn the appropriate coupling term from human demonstrations of obstacle avoidance. Our

results demonstrate that human-like obstacle avoidance can be accomplished from reactive feedback terms. We performed experiments on the simulated version of Apollo, depicted in Figure 1.1, as well as several experiments in simulated MATLAB environments.

CHAPTER 2

RELATED WORK ON OBSTACLE AVOIDANCE

Obstacle avoidance is a widely addressed problem in robotics and still remains unsolved. Most methods that address it in literature can be broadly classified into two categories- local and global methods. Local methods use the obstacle and its relationship with the robot locally to control, such as the Vector-Field Histogram [2], curvature velocity method [19] and motion field flow [12]. These local methods are reactive and fast, but might lead to sub-optimal solutions as they do not take the whole environment into account. Global obstacle avoidance methods are path planning methods like [10], which have a global information of the obstacles and the environment. Others like the certainty grid methods in [11] take a probabilistic approach of obstacle locations in the environment to take into account inexact sensor data while path planning. These methods can find an optimal solution, if it exists, even in very complex environments but are often computationally expensive and, thus, less reactive to suddenly appearing obstacles. There have been attempts to optimize these methods in [3], but they are still less reactive than local methods.

A more practical approach are hybrid systems, that switch between global and local methods. The solution may not be optimal, and might still suffer from the drawbacks of a local algorithm. Some methods like [1] break the task into small tasks that are solvable locally. If this decomposition fails, global planning is invoked.

Reshaping methods like the elastic band approach [17] provide another way to fill the gap between global path planning and reactive sensor-based control. The elastic band is a collision-free path, that starts from a path generated by a planner and that is deformed when encountering new obstacles. However, if the original

path becomes infeasible due to changes in the environment, this method is unable to find a new feasible path. A reactive method proposed by [22] mixes path re-planning and deformation using efficient roadmap use and parallel planning and execution. Thus when local deformation no longer works and the original path becomes infeasible, we can re-plan.

In artificial potential field approaches – static or time-varying – as proposed by [9] potential fields are built in the environment around the robot, with attractive fields assigned to the goal and repulsive fields to the obstacles. These can be used to avoid obstacles by not just the end-effector but all of the manipulator links. However, these are difficult to implement in complex environments and the robot can get stuck in a local minima even if a path to the goal actually exists. In the attractor dynamics approach, as described in [7], the heading of the end-effector is used instead of the position, that helps escape local minima points that hamper potential field methods.

Potential fields with DMPs were used in [13] to avoid point obstacles. This method can easily handle multiple obstacles. Furthermore, it ensures obstacle avoidance for the whole manipulator link and not just the end-effector. However, it does not deal with extended obstacles and it can be computationally expensive to calculate potential fields on the fly. Other optimization based obstacle avoidance methods, as described in [18] and [8] can be used in combination with DMPs. These are however useful in static or quasi-static environments. A reactive method of obstacle avoidance with DMPs was described in [14] to avoid obstacles with the help of direct feedback from the environment. This method uses coupling terms in DMPs as a local obstacle avoidance function that uses the heading direction and distance from obstacle to avoid moving point obstacles. However, it does not take extended obstacles into account.

With these ideas in mind, we use a DMP, which is a pre-planned path for manipulation and modify our original plan using coupling terms. Unlike earlier works, our method is reactive and capable of avoiding a wide range of obstacles in a human-like fashion. These coupling terms consist of local features like distance from obstacle, velocity of end-effector, etc. to avoid extended obstacles.

In the following chapters, we describe the DMP formulation, followed by our coupling term formulation. We then describe the learning methods employed to make our coupling term robust and finally, show experimental results for our formulation.

CHAPTER 3

BACKGROUND ON DYNAMIC MOVEMENT PRIMITIVES

DMPs consist of a simple point attractive model, instantiated by the second order dynamics given in Eq. 3.1. There are three basic components of a DMP: the transformation system, the non-linear function and the canonical system. The *transformation system* of the DMP is given by

$$\tau\ddot{y} = \alpha_z(\beta_z(g - y) - \dot{y}) + f + C_t \quad (3.1)$$

where g is the known goal position, α_z and β_z are time constants, τ is a temporal scaling factor and y , \dot{y} , \ddot{y} are the position, velocity and acceleration. f is a non-linear term that allows modelling of almost arbitrarily complex movements by the DMP. C_t is called the spatial coupling term that can be used to modify the DMP online.

Eq. 3.1 can be interpreted as a spring-mass system with added non-linear terms f and C_t . For $f = 0$ and $C_t = 0$ this system forms a globally stable system that converges exponentially to the attractor point given by $(y, \dot{y}, \ddot{y}) = (g, 0, 0)$, which means that starting from any arbitrary point, the system converges to the goal g . These non-linear terms give the system more complex dynamics than the trivial convergence to g and can be modelled using function approximators, such as weighted exponential kernels.

The non-linear function f can most easily be defined as an explicit function of time. However, explicit time dependence inhibits a straight forward mechanism to couple and coordinate the DMP with other dynamical systems. Thus the DMP framework uses an additional dynamic system that maintains the phase x of the movement to avoid explicit time dependencies.

This system is called the *canonical system* which is among the most basic dynamic systems available with a point attractor at $x = 0$.

$$\tau \dot{x} = -\alpha_x x + C_c \quad (3.2)$$

x is a monotonically decreasing variable that goes from 1 to 0 in a time period characterized by τ . C_c is a coupling term that can be used to modulate the temporal evolution of the system.

Subsequently, f can be modelled as a function of x , using, for example, exponential kernels:

$$f(x, g, y_0) = \frac{\sum_{i=1}^N \psi_i w_i}{\sum_{i=1}^N \psi_i} x \quad (3.3a)$$

$$\psi_i = \exp(-h_i(x - c_i)^2) \quad (3.3b)$$

The influence of f vanishes after x has decayed to 0.

3.1 Modelling and generating movement with DMPs

A DMP can be initialized from a demonstration that provides a trajectory consisting of positions $y_{demo}(t)$, velocities $\dot{y}_{demo}(t)$ and accelerations $\ddot{y}_{demo}(t)$. From this information, the non-linear function Eq. 3.3a can be regressed using the targets

$$f_{target}(s) = \tau \ddot{y}_{demo} - \alpha_z (\beta_z (g - y_{demo}) - \dot{y}_{demo}) \quad (3.4)$$

The resulting weights w_i minimize the error function $J = \sum_s (f_{target}(s) - f(s))^2$. After learning the weights, a movement is generated as follows. The transformation system of Eq. 3.1 is at rest when $y = g$ and $\dot{y} = 0, \ddot{y} = 0$. To trigger a movement, x is set to 1, $y = y_0, g = g_{desired}$. The desired attractor landscape is obtained by plugging in the learnt w_i into f and the desired movement duration is set to τ . The canonical system is integrated to give $x(t)$ and the corresponding f is calculated, followed by integration of the transformation system to generate the next required position and velocity.

This representation of a movement is robust to changes in goal, initial point,

duration and scale of the movement. It can also be modulated online using the spatial and temporal coupling terms mentioned before in Eq 3.1 and 3.2 and as described in the next section.

3.2 Coupling terms

Coupling terms in DMPs can be used to couple sensory information with the control of a robot in the canonical systems as well as the transformation system of the DMP. For example, previously stored information from a force sensor was used to grasp an object with uncertainty in position and orientation in [16] by modulating the transformation system. Coupling terms can also be used to modulate DMPs so as to not move beyond joint-angle limits [4]. In [5], the authors proposed a way to learn coupling terms based on sensor observations during task executions and used this as a feedback when executing the same task. In this way, DMPs perform coupled bi-manual tasks efficiently by learning the external force arising from the interaction. Learnt coupling terms from previous executions of a task can also be used to associate sensory information with the task, as proposed in [15]. Particular features can become active in each situation leading to a particular kind of behaviour.

CHAPTER 4

COUPLING TERMS FOR OBSTACLE AVOIDANCE

[21] proposed a human-inspired obstacle avoidance function for point obstacles in 2D which was developed by [14] into a specialized coupling term C_t that could be added to the transformation system of a DMP to avoid point obstacles.

In 2D, as shown for human walking experiments in [21], a simple obstacle avoidance function can be given as

$$\dot{\theta} = \theta \exp(-\beta\theta) \quad (4.1a)$$

where, θ is the angle between the velocity and the vector joining the end-effector with the obstacle, as shown in Figure 4.1. To transform this into 3D Cartesian coordinates, express $\dot{\mathbf{y}}$ as $[\dot{\mathbf{y}} \cos(\theta), \dot{\mathbf{y}} \sin(\theta)]$ and differentiate it with respect to time:

$$\begin{aligned} \dot{\mathbf{y}} &= [|\dot{\mathbf{y}}| \cos(\theta), |\dot{\mathbf{y}}| \sin(\theta)] \\ \frac{d\dot{\mathbf{y}}}{dt} &= \frac{d\dot{\mathbf{y}}}{d\theta} \frac{d\theta}{dt} = [-|\dot{\mathbf{y}}| \sin(\theta) \frac{d\theta}{dt}, |\dot{\mathbf{y}}| \cos(\theta) \frac{d\theta}{dt}] \\ &= [\dot{\mathbf{y}} \cos(\theta + \pi/2), \dot{\mathbf{y}} \sin(\theta + \pi/2)] \frac{d\theta}{dt} \end{aligned}$$

$[\dot{\mathbf{y}} \cos(\theta + \pi/2), \dot{\mathbf{y}} \sin(\theta + \pi/2)]$ is $\dot{\mathbf{y}}$ rotated by $\pi/2$. Thus, we can write $\ddot{\mathbf{y}}$ as

$$\ddot{\mathbf{y}} = R\dot{\mathbf{y}}\dot{\theta} \quad (4.1b)$$

$$= R\dot{\mathbf{y}}\theta \exp(-\beta\theta) \quad (4.1c)$$

$$(4.1d)$$

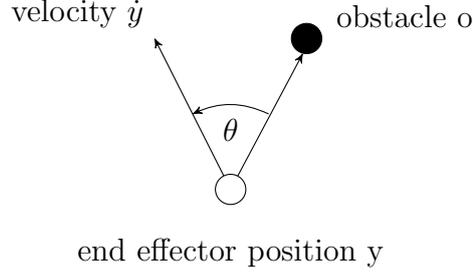


Figure 4.1: The steering angle in the plane of the obstacle, end-effector and the direction of motion

Since C_t can be seen as a repulsive force away from the obstacle,

$$C_t = \gamma \ddot{\mathbf{y}} \quad (4.1e)$$

$$C_t = \gamma \mathbf{R} \dot{\mathbf{y}} \theta \exp(-\beta \theta) \quad (4.1f)$$

where

$$\theta = \cos^{-1} \left(\frac{(\mathbf{o} - \mathbf{y})^T \dot{\mathbf{y}}}{|(\mathbf{o} - \mathbf{y})| |\dot{\mathbf{y}}|} \right)$$

$$\mathbf{R} = \text{RotationMatrix}(\mathbf{r}, \pi/2)$$

$$\mathbf{r} = (\mathbf{o} - \mathbf{y}) \times \dot{\mathbf{y}}$$

Here, \mathbf{o} is the obstacle position and \mathbf{y} and $\dot{\mathbf{y}}$ are the end-effector position and velocity respectively. This 3D coupling term returns a (3×1) vector that can be seen as a repulsive force perpendicular to $\dot{\mathbf{y}}$ away from the obstacle, in the plane containing the obstacle, end-effector and velocity. $\mathbf{R} \dot{\mathbf{y}}$ tries to move the heading of the end-effector by $\pi/2$ while $\theta \exp(-\beta \theta)$ makes sure that the end-effector is repulsed when its moving towards the obstacle. Thus, if the end-effector of a robot moves away from the obstacle, it doesn't experience any repulsive force. However, there are a few problems with this formulation of obstacle avoidance, as described below.

1. It does not ignore far-away obstacles, which would be expected from a robot.
2. This function is 0 for $\theta = 0$, which means the force is 0 when directly heading towards the obstacle.

3. This might fail to avoid extended obstacles: This formulation mainly takes the obstacle center into account. For large obstacles, a little more information about the geometry of the obstacle might be useful.

To counter the above mentioned shortcomings, we modify this original coupling term as described below:

1. To diminish the effect of distant obstacles we multiply the original coupling term by a negative exponential of the distance.
2. To avoid 0 coupling term when moving directly towards the obstacle, we added another component to the coupling term which is a spherical field of repulsion away from the obstacle.
3. We use one or more points on the surface of the obstacle for obstacle avoidance. However, this leaves us with the problem of choosing such a point. One obvious point on the surface of the obstacle could be the closest point on the surface of the obstacle. However it can be complex to calculate this without the knowledge of the geometry of the obstacle. Since, we are dealing with the plane passing through the obstacle center and end-effector, one easy way to approximate the nearest point is to find the intersection of the vector from end-effector to obstacle center and the obstacle boundary (Figure 4.2). However, instead of choosing this particular point, we can also choose several points on the obstacle surface, for example corners or midpoints of edges, that seem to be representative of the obstacle.

It should be noted that our coupling term is still a local fashion of obstacle avoidance that doesn't need the overall geometry of the obstacle.

This gives three components to our new coupling term, which in 2D can be written as

$$\phi_1 = \theta \exp(-\beta\theta) \exp(-kd) \tag{4.1g}$$

$$\phi_2 = \theta \exp(-\beta\theta_p) \exp(-kd_p) \tag{4.1h}$$

$$\phi_3 = \exp(-kd) \tag{4.1i}$$

where d is the distance of the end-effector from the center of the obstacle and d_p is the distance from the nearest point. The function $\theta \exp(-\beta\theta)$ has a behaviour

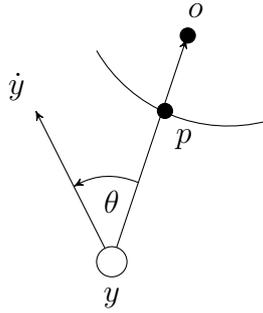
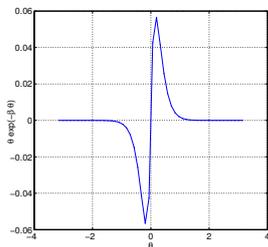
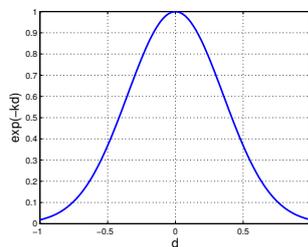


Figure 4.2: Plane defined by the obstacle center \mathbf{o} , end-effector position \mathbf{y} and velocity $\dot{\mathbf{y}}$. The ‘closest’ point \mathbf{p} is the point which lies in this plane, and is defined by the intersection of the vector joining \mathbf{o} and \mathbf{y} and the obstacle surface.

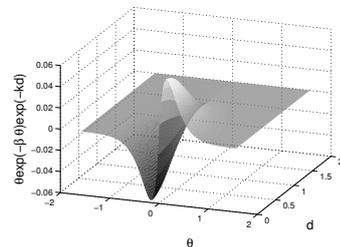
as shown in Figure 4.3a. $\exp(-kd)$ is the negative exponential of the distance that diminishes the effect of far-away obstacles. The combined effect of these two terms can be seen in Figure 4.3c. ϕ_1 can be seen as a force that is directed away from the center of the obstacle and ϕ_2 is a repulsive force away from the nearest point on the obstacle. ϕ_3 is a spherical field around the obstacle that reduces exponentially with the distance, as shown in Figure 4.3b.



(a) $\theta \exp(-\beta\theta)$ as a function of θ



(b) $\exp(-kd)$ as a function of d



(c) $\theta \exp(-\beta\theta) \exp(-kd)$ as a function of θ and d

Figure 4.3: Different constituents of the features in (a),(b) 2D, (c) 3D

CHAPTER 5

LEARNING COUPLING TERM

Based on the features mentioned in the last chapter, we can now formulate our coupling term as a weighted linear combination of three terms

$$C_t = \gamma_1\phi_1 + \gamma_2\phi_2 + \gamma_3\phi_3 \quad (5.1a)$$

where ϕ_1 and ϕ_2 can be seen as repulsive forces away from the obstacle center and a point on the surface of the obstacle. ϕ_3 is a uniform exponential field pointing away from the obstacle center.

These features, when transformed into a Cartesian coordinate frame and combined linearly, give rise to a coupling term that seems to be capable of avoiding extended obstacles at almost any angle.

$$C_t = \gamma_1\mathbf{R}\dot{\mathbf{y}}\phi_1 + \gamma_2\mathbf{R}\dot{\mathbf{y}}\phi_2 + \gamma_3\mathbf{R}\dot{\mathbf{y}}\phi_3 \quad (5.1b)$$

Some experiments with this coupling term and hand-tuned parameters γ, k, β can be seen in Chapter 7. In case of multiple obstacles, a weighted linear combination of the C_{ti} from all the obstacles o_i

$$C_{tTotal} = \sum_{i=1}^N w_i C_{ti} \quad (5.1c)$$

can be used. Obstacles can, for example, be weighted according to their proximity to the goal. Those that are close to the goal get a lower weight than those that are farther away, to allow the DMP to still reach the goal, even when crowded by different obstacles.

5.1 Basis of features

To be able to avoid obstacles in a wide range of situations and environments, we need to extend the features described above to cover a larger basis. A closer look at the features helps us to analyze the significance of the constituting components and create a more general basis.

5.1.1 ϕ_1 and ϕ_2

$$\phi_i = \theta_i \exp(-\beta\theta_i) \exp(-kd_i)$$

Here θ_i , and d_i are as explained earlier in Figure 4.1 and 4.2. These can be calculated by choosing appropriate points on the obstacle for obstacle avoidance.

β and k can, however, be seen as hyper-parameters that depend on the obstacle and DMP. The peak of the curve shown in Figure 4.3a is at $1/\beta$. Higher β implies a very narrow region of influence of the obstacle. This means that the obstacle repels the end effector only when it moves towards the obstacle. On the other hand, a higher k means that the effect of the obstacle decreases very quickly with distance. Thus, the obstacle repulsion only comes into picture when the end-effector comes very close to the obstacle, as shown in Figure 4.3. In 3D, after combining the two, this feature can be visualized as shown in Figure 4.3

Typically, we would like higher β and k if we are either moving very slowly, which gives us time to react, or if the obstacle is very small. On the other hand, for a fast moving trajectory and big, elongated obstacles, we would like a smaller β and k . However, very large β can cause peaks in the coupling term, thus we use only moderately high values. Also, we might not want to consider the points on the boundary in the case of small obstacles, but give them more importance when dealing with larger obstacles.

Thus, different obstacles would need different parameter values to be avoided effectively.

5.1.2 ϕ_3

$$\phi_3 = \exp(-kd)$$

This term uses only the distance from the obstacle and does not take into account the heading of the end-effector. k determines the range of influence of each obstacle. Ideally, we would like obstacles that are not in the path of the end-effector to have lower influence than those that come in the path directly. Also, larger obstacles should have a larger range than smaller obstacles.

As a solution we propose to generate a general basis spanning over intervals of the parameters. The simplest basis of features could be created on a grid. An example grid could be:

$$k := \{1, 2, 3, 4, 5\}$$
$$\beta := \{5/\pi, 10/\pi, 15/\pi, 20/\pi, 25/\pi\}$$

while using just two points on the obstacle - center and nearest point.

This approach would give us a wide range of features that should be capable of creating coupling terms for obstacles avoidance for a larger variety of obstacles. One could also consider including more points on the obstacle as added features to the basis, for example, a grid of points around the closest point. Finally, we need to determine the relevant features within that basis and the optimal parameters γ to combine them to form the coupling term C_t . Given some target values, this can be achieved using linear regression with automatic relevance determination.

5.2 Linear Regression to learn the coupling terms

A first step is to learn the weights γ introduced in Equation 5.1b. Since C_t is linear in γ , this can be solved using linear regression. The target values of C_t can be extracted from simulation experiments, robot demonstrations or human demonstrations. Here, we used human demonstrations to calculate our target coupling term, as described in Chapter 7 to keep the behaviour of our DMPs as close to human-like obstacle avoidance as possible. Let our target C_t over N

observations be represented as \mathbf{t} , then we have the following relationship

$$\mathbf{t} = \sum_i \gamma_i R \dot{\mathbf{y}} \phi_i = \mathbf{\Phi} \boldsymbol{\gamma} \quad (5.4a)$$

where, $\boldsymbol{\gamma}$ is the vector of γ_i and $\mathbf{\Phi}$ is the matrix formed by $R \dot{\mathbf{y}} \phi_i$ for N observations. Solving for $\boldsymbol{\gamma}$ with regularization gives

$$\boldsymbol{\gamma} = (\alpha \mathbf{I} + \mathbf{\Phi}^T \mathbf{\Phi})^{-1} \mathbf{\Phi}^T \mathbf{t} \quad (5.4b)$$

For more freedom in fitting our model from the data, we take the different spatial dimensions of our coupling term to be independent and fit three separate regression models on the x,y and z dimensions, while using features from all dimensions. This triples the total number of parameters in our model, giving it more freedom to fit an arbitrary function.

5.3 Automatic Relevance Determination

The set of basis functions created on a grid might have a lot of features that would actually prove unnecessary for most obstacles. Or they may not contribute any new information to the coupling term. Keeping the whole basis makes the algorithm slow as these features have to be calculated at each step. Thus, we apply the idea of Bayesian regression with automatic relevance determination [20] on our basis to prune out unnecessary features.

The likelihood of the targets \mathbf{t} is given by

$$p(\mathbf{t} | \mathbf{\Phi}, \boldsymbol{\gamma}, \beta) = \prod_{n=1}^N \mathcal{N}(t_n | \phi_n \boldsymbol{\gamma}, \beta^{-1}) \quad (5.5)$$

We introduce a factorizing prior over $\boldsymbol{\gamma}$

$$p(\boldsymbol{\gamma} | \boldsymbol{\alpha}) = \prod_{i=1}^M \mathcal{N}(\gamma_i | 0, \alpha_i^{-1}) \quad (5.6)$$

which will allow us to perform automatic relevance determination.

The posterior distribution is proportional to the product of the likelihood and the

prior. As both the distributions are Gaussian, the posterior will also be Gaussian. Using results for marginal and conditional distribution on Gaussian distribution, we get the posterior over $\boldsymbol{\gamma}$ as

$$\begin{aligned} p(\boldsymbol{\gamma}|\mathbf{t}, \boldsymbol{\Phi}, \boldsymbol{\alpha}, \beta) &= \mathcal{N}(\boldsymbol{\gamma}|\mathbf{m}, \boldsymbol{\Sigma}) & (5.7) \\ \mathbf{m} &= \beta \boldsymbol{\Sigma} \boldsymbol{\Phi}^T \mathbf{t} \\ \boldsymbol{\Sigma} &= (\mathbf{A} + \beta \boldsymbol{\Phi}^T \boldsymbol{\Phi})^{-1} \end{aligned}$$

where $\mathbf{A} = \text{diag}(\alpha_i)$. The values of $\boldsymbol{\alpha}$ and β are determined by type-2 maximum likelihood, also known as *evidence approximation*. This maximizes the maximum likelihood function obtained by integrating out the weight parameters

$$p(\mathbf{t}|\boldsymbol{\Phi}, \boldsymbol{\alpha}, \beta) = \int p(\mathbf{t}|\boldsymbol{\Phi}, \boldsymbol{\gamma}, \beta) p(\boldsymbol{\gamma}|\boldsymbol{\alpha}) d\boldsymbol{\gamma} \quad (5.8)$$

Because this is the convolution of two Gaussians, it can be evaluated to give the log marginal likelihood as

$$\log p(\mathbf{t}|\boldsymbol{\Phi}, \boldsymbol{\gamma}, \beta) = \log \mathcal{N}(\mathbf{t}|\mathbf{0}, \mathbf{C}) \quad (5.9)$$

$$= -\frac{1}{2} N \log(2\pi) + \log |\mathbf{C}| + \mathbf{t}^T \mathbf{C}^{-1} \mathbf{t} \quad (5.10)$$

where

$$\mathbf{C} = \beta^{-1} \mathbf{I} + \boldsymbol{\Phi} \mathbf{A}^{-1} \boldsymbol{\Phi}^T \quad (5.11)$$

Our aim is to maximize the log-likelihood with respect to $\boldsymbol{\alpha}$ and β . By setting the derivatives of the marginal likelihood to zero, we obtain the following estimates

$$\alpha_i^{\text{new}} = \frac{\lambda_i}{m_i^2} \quad (5.12a)$$

$$(\beta^{\text{new}})^{-1} = \frac{\|\mathbf{t} - \boldsymbol{\Phi} \mathbf{m}\|^2}{N - \sum_i \lambda_i} \quad (5.12b)$$

$$\lambda_i = 1 - \alpha_i \Sigma_{ii} \quad (5.12c)$$

where m_i is the i -th component of \mathbf{m} and Σ_{ii} is the i -th diagonal element of $\boldsymbol{\Sigma}$. We start with an initial value of β and $\boldsymbol{\alpha}$ and iteratively update \mathbf{m} , $\boldsymbol{\Sigma}$ and then β

and $\boldsymbol{\alpha}$ until a convergence criteria is reached.

As we maximize the log-likelihood iteratively, most of α_i go to infinity, making the corresponding weights γ_i have posterior distributions concentrated at zero. The basis functions associated with these weights, therefore, play no role in the prediction and are effectively pruned out, resulting in a sparse model.

CHAPTER 6

CONVERGENCE TO GOAL

Adding coupling terms to a DMP raises concerns over its ability to converge to the goal of the movement. One could expect that if the obstacle is very close to the goal, it might get tougher for the DMP to reach the goal. For a stationary obstacle, we can prove that our formulation of the coupling term satisfies the Lyapunov stability criterion and converges to the goal for all starting points.

As $t \rightarrow \infty$ phase variable x goes to 0. Thus, the reduced DMP equation is

$$\tau \ddot{\mathbf{y}} = \alpha_z (\beta_z (\mathbf{g} - \mathbf{y}) - \dot{\mathbf{y}}) + C_t \quad (6.1a)$$

This can be simplified as

$$\ddot{\mathbf{y}} = K(\mathbf{g} - \mathbf{y}) - D\dot{\mathbf{y}} + C_t \quad (6.1b)$$

$$C_t = \gamma_1 \Phi_1 + \gamma_2 \Phi_2 + \gamma_3 \Phi_3 \quad (6.1c)$$

where γ_i is the vector of weights γ and Φ_i is the matrix of the features ϕ_i , as described before.

The point $(\mathbf{y}, \dot{\mathbf{y}}, \ddot{\mathbf{y}}) = (\mathbf{g}, 0, 0)$ is a stationary point for this equation. We can construct a Lyapunov function V and prove $\dot{V} < 0 \forall (\mathbf{y}, \dot{\mathbf{y}}, \ddot{\mathbf{y}}) \neq (\mathbf{g}, 0, 0)$, hence showing that all starting points of the system will converge to this stationary point.

We use the energy function of a damped spring mass system $\ddot{\mathbf{y}} = K(\mathbf{g} - \mathbf{y}) - D\dot{\mathbf{y}}$

$$V(y, \dot{\mathbf{y}}, \ddot{\mathbf{y}}) = \frac{1}{2}(\mathbf{g} - \mathbf{y})^T K(\mathbf{g} - \mathbf{y}) + \frac{1}{2}\dot{\mathbf{y}}^T \dot{\mathbf{y}} \quad (6.1d)$$

$$\dot{V} = \Delta_{\mathbf{y}} V^T \dot{\mathbf{y}} + \Delta_{\dot{\mathbf{y}}} V^T \ddot{\mathbf{y}} \quad (6.1e)$$

$$= -(\mathbf{g} - \mathbf{y})^T K \dot{\mathbf{y}} + \dot{\mathbf{y}}^T \ddot{\mathbf{y}} \quad (6.1f)$$

$$(6.1g)$$

From Equation 6.1c,

$$= -\dot{\mathbf{y}}^T D \dot{\mathbf{y}} + \dot{\mathbf{y}}^T C_t \quad (6.1h)$$

$$= -\dot{\mathbf{y}}^T D \dot{\mathbf{y}} \quad (6.1i)$$

as $\dot{\mathbf{y}}$ and $R\dot{\mathbf{y}}$ are rotated by $\pi/2$, making $\dot{\mathbf{y}}^T C_t = 0$. This proves that the DMP would converge to the stationary point of $(\mathbf{y}, \dot{\mathbf{y}}, \ddot{\mathbf{y}}) = (\mathbf{g}, 0, 0)$ from any starting point.

EXPERIMENTS

We evaluate our coupling term formulation in several ways. We start with a qualitative discussion of the ability of our coupling term to avoid obstacles reactively, followed by its ability to model different types of user demonstrations, per subject, across subjects and across obstacles. Then we move on to a description of the ability of the thus learnt coupling term to avoid obstacles. We also test our coupling term on movements of different durations.

First, we show the effectiveness of the new coupling term to avoid obstacles in simulated settings, with hand-tuned parameters.

7.1 Evaluation of the coupling term on simulated examples

To see the ability of our coupling term formulation to avoid extended and multiple obstacles, we conducted several experiments on trajectories generated in simulation. In order to do so, we created initial trajectories and fitted a DMP for them, using Equation 3.4. Then, we equipped the initial DMP with a coupling term $C_{\text{tuned}} = \gamma_1\phi_1 + \gamma_2\phi_2 + \gamma_3\phi_3$ using our new features with hand-tuned parameters $\tilde{k}, \tilde{\beta}$ and hand tuned weights $\tilde{\gamma}_1, \tilde{\gamma}_2, \tilde{\gamma}_3$ and integrated it. While unrolling the DMP, we progressively added obstacles in the path. An example of this is shown in Figure 7.1. This example shows the reactive nature of the coupling term. The obstacles appear suddenly as the DMP (with active coupling term) is unrolling. Before any obstacle was present, the DMP follows the same path as the initial trajectory. As obstacles appear, the coupling term changes the initial DMP path

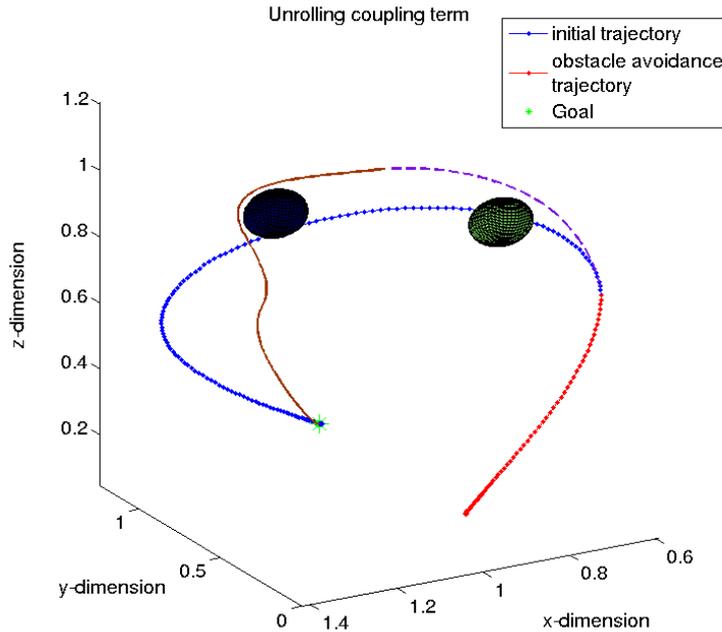


Figure 7.1: DMP motion generation with obstacles and coupling terms. Obstacle appears after start of the motion. Until then, the obstacle avoidance trajectory (red) is the same as the initial trajectory (blue). Obstacle 1 is avoided (purple) but quickly afterwards, obstacle 2 appears. Obstacle 2 is avoided (brown) and the trajectory converges to the goal.

to successfully avoid the two obstacles. Furthermore the DMP converges to the goal, though it takes longer.

It should be noted that this experiment required hand-tuned parameters. If the parameters were kept constant, but the size, position or time of appearance of obstacle changed, the coupling term may no longer be able to avoid the obstacle. This, again pointed towards the need for a more general basis of features.

So, next we show that the parameters γ for a general basis of features Φ , as described in Chapter 5, can be learned to reconstruct coupling terms extracted from human demonstrations. This means that with a more powerful coupling term we were able to avoid obstacles in a human-like fashion.

7.2 Coupling term from human demonstrations

We collected human demonstrations of basic obstacle avoidance behaviors on two experimental set ups. The goal of this experiment was two-fold: First, we



Figure 7.2: The Vicon setup showing the start, goal points, cylindrical obstacle and a human demonstrator guiding a set of markers through the environment.

wanted to verify that our coupling term formulation can represent coupling terms extracted from human demonstrations. Second, learning parameters γ for an overcomplete set of features. This will determine which features contribute towards a general coupling term parametrization that can generate human-like obstacle avoidance behaviour within the DMP framework.

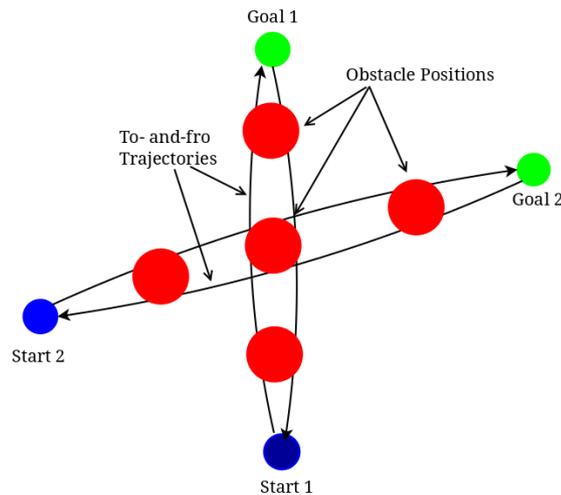
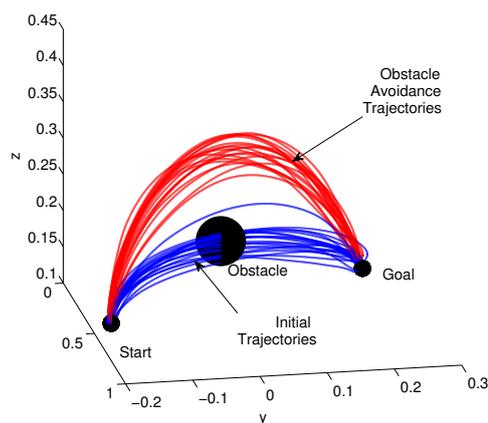


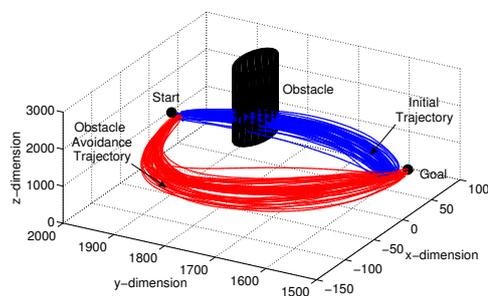
Figure 7.3: A 2-D representation of the experimental set-up showing the to-and-fro paths between start (blue) and goal (green) points, as well as the obstacle locations (red). Note that the subject position is fixed, and hence the two paths need quite different movements by the subject. Also, data to the goal from the start point, as well as, from the goal to the start was recorded and analyzed.

To collect demonstrations, we used a 3D Guidance trakSTAR system that measures the position of a magnetic sensor in a magnetic field set up by a transmitter. Additionally, we collected data on a Vicon 3D Motion Capture system. The Vicon

setup can be seen in Figure 7.2. In both set ups, subjects were asked to reach with their hands from an initial position to a goal position, first without and then with an obstacle in the path. Three types of obstacles - sphere, cylinder and ellipsoid, placed at three different locations on two different paths were used. A 2-D representation of our start, goal and obstacle positions can be seen in Figure 7.3.



(a)



(b)

Figure 7.4: Demonstrations in 3D with and without (a) a spherical obstacle and (b) a cylindrical obstacle

On the trakSTAR set up, 7 subjects demonstrated 100 trajectories for both initial and obstacle avoidance trajectories. On the Vicon set up we collected data from 40 subjects, demonstrating 30 movements in each phase. A subset of the demonstrated trajectories of one subject for a spherical (collected using trakStar) and cylindrical (collected using Vicon) obstacle are shown in Figure 7.4.

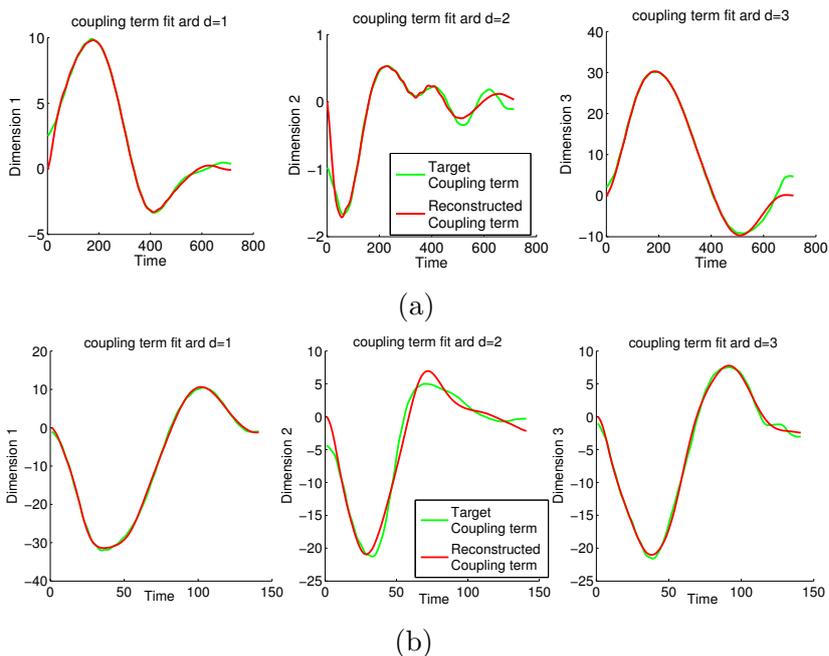


Figure 7.5: Dimensions 1, 2 and 3 of the target and reconstructed coupling term using a basis of features, as calculated using regression with ARD for (a) Sphere and (b) Cylinder. The fits are quite close to human demonstrations, showing that our coupling term formulation is capable of reproducing human like behaviour. Notice that the two targets are quite different in the first and third dimensions, but the features are able to capture this difference well.

7.3 Modelling human behaviour

To fit this target coupling term collected from human demonstrations, we create a basis of features as described in Chapter 5, with a total of 615 features. Finally, we use Bayesian linear regression with automatic relevance determination, to find the optimal set of parameters γ to reconstruct the human-like target coupling term C_t . Representative results of the fit for the three dimensions of our coupling term on the human demonstrations can be seen in Figure 7.5. These results indicate that our features can indeed represent coupling terms extracted from human obstacle avoidance movements. The upper row shows the coupling term fit, when learning across all subjects on the trakStar data. Although subjects displayed slight variations in avoidance on this set-up, our coupling term features were able to capture the general behaviour well. Similarly, the lower row shows the resulting coupling term fit on the data collected on the Vicon system.

7.3.1 Limitations in modelling different behaviours

Subjects differ in their obstacle avoidance behaviours. We asked them to follow a specific strategy in one experiment, but this strategy could be different from subject to subject. Also, different obstacles need different strategies to be avoided. For example, when avoiding a tall cylinder, most subjects move around it to reach the goal, while when avoiding a sphere, they tend to move on top of it. Ellipsoids seem to have the maximum variance across subjects, with some moving over them, while others moving around them. Even for the same obstacle, subjects could follow different strategies for the two paths shown in Figure 7.3 as they need quite different movements of the arm.

Even though these seem to be similar situations, they lead to very different target coupling terms to be fit for the same environment. This can be seen in Figure 7.5. The coupling terms are different in scale and shape for all the dimensions, for the same obstacle position and initial path. Subjects deviate mostly only in z-direction for a sphere, where as in almost all three dimensions for a cylinder. Ideally, we would want our coupling term to be independent of these factors and be able to model all of these movements.

Interestingly, our coupling term is able to model each of these behaviours individually very robustly. This means that our features are informative enough to extract this information from the demonstration well. Even across to and fro movements on the same trajectory (i.e., from start to goal and goal to start), the coupling term fits the behaviour well. However, if we try to fit our coupling term across different behaviours, by using target coupling term values for multiple subjects with different obstacle avoidance behaviour, we find our coupling term does not fit our target values so well.

This can be explained by the fact that these behaviours are basically different ‘modes’ of obstacle avoidance. For example, for one demonstration the coupling term could have a large influence in the y-dimension while for the other in the z-dimension. When trying to fit across these different modes, linear regression tries to average out the two behaviours as the features are nearly the same for the start of any movement (approximately 0). In such a scenario, the coupling term fits the most consistent dimension, across demonstrations well, while the dimensions that have a lot of variance across subjects are not fit that well. However, when we consider back and forth movements on the same path, the ‘mode’ of obstacle

avoidance remains the same, but the target C_t is inverted in one or more dimension. As a result, our coupling term is able to capture these movements together. Figure 7.6 and 7.7 illustrate these points well for an ellipsoidal obstacle, which has a large amount of variance amongst subjects.

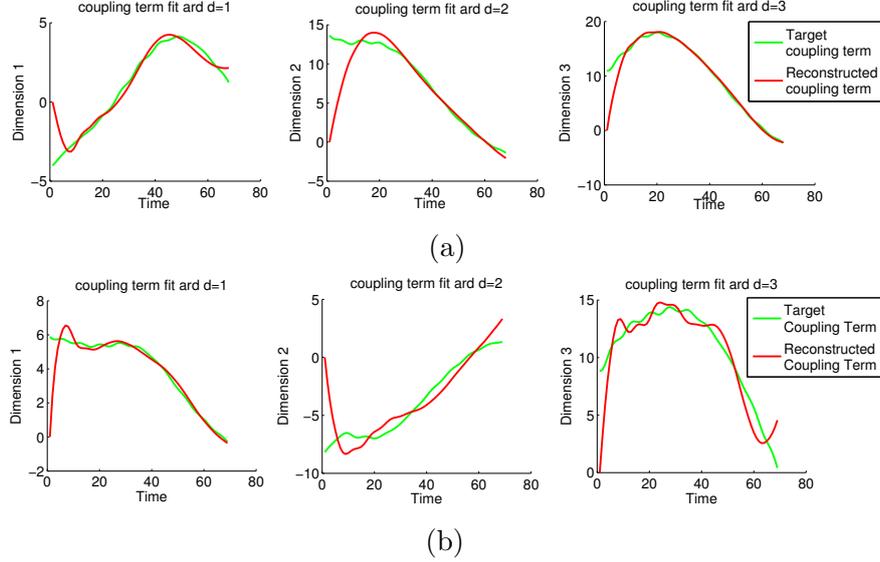


Figure 7.6: Dimensions 1, 2 and 3 of the target and reconstructed coupling term using a basis of features, as calculated using regression with ARD across (a) path from goal to start and (b) path from start to goal for an ellipsoid. The fits are quite close to human demonstrations, showing that our coupling term formulation is capable of reproducing human like behaviour from the same ‘mode’ of obstacle avoidance. Notice how the coupling terms in Dimensions 1 and 2 mirror each other.

This could point to the need of an additional hidden variable that represents the mode of obstacle avoidance when learning the coupling term.

7.4 Unrolling coupling term from human demonstrations

A final step is to use the learnt parameters $\hat{\gamma}$ to unroll the initially learnt DMP now equipped with coupling terms. In order to do so, we use the initial DMP with an added coupling term, such that the *transformation system* of the DMP is given by

$$\tau \ddot{y} = \alpha_z (\beta_z (g - y) - \dot{y}) + \hat{f}_{\text{initial}} + \phi \hat{\gamma} \quad (7.1)$$

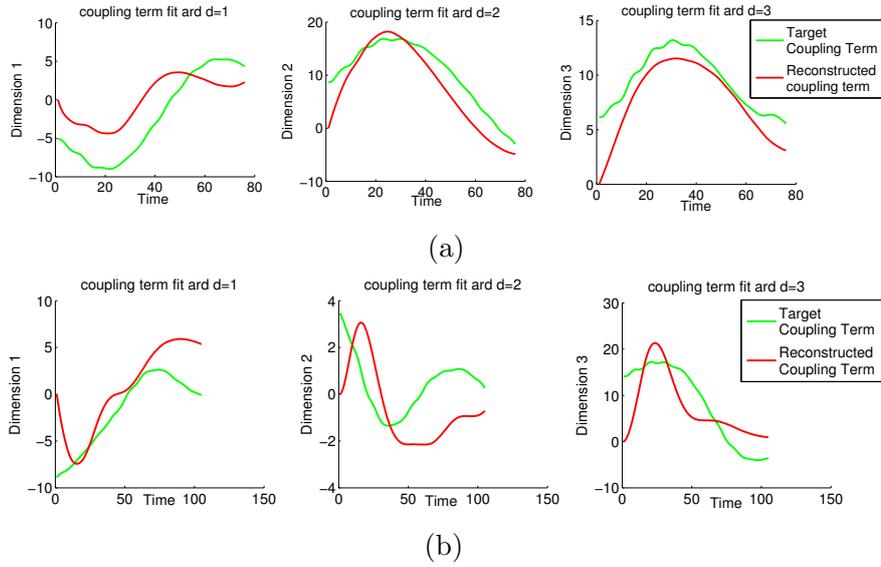


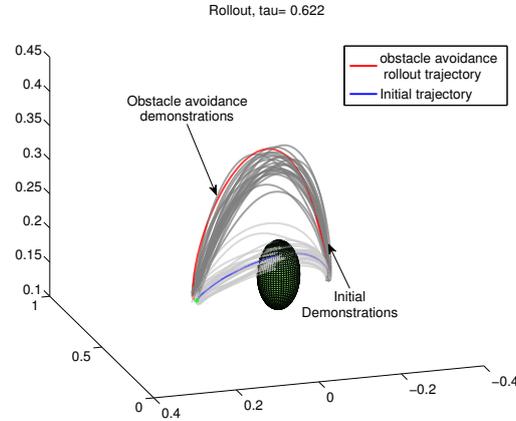
Figure 7.7: Dimensions 1, 2 and 3 of the target and reconstructed coupling term using a basis of features, as calculated using regression with ARD across (a) first mode and (b) second mode for an ellipsoid. The fits are not that good anymore, showing that the coupling term doesn't generalize well across different behaviours. Notice how (a) avoids the obstacle by more movement in y-dimension, as seen by higher coupling term while (b) does so by movement in z-dimension.

where the features ϕ are computed on the fly and $\hat{\gamma}$ were learnt from human demonstrations. Note that in this setting, the output of the coupling term model determines the next step of the DMP, which in turn influences the input of the model in the next step. Thus, if the DMP starts to deviate from its path to a region not generalized well by the model, this forms a feedback loop, causing the divergence to increase. It is therefore important to use regression models that generalize well to even unknown areas.

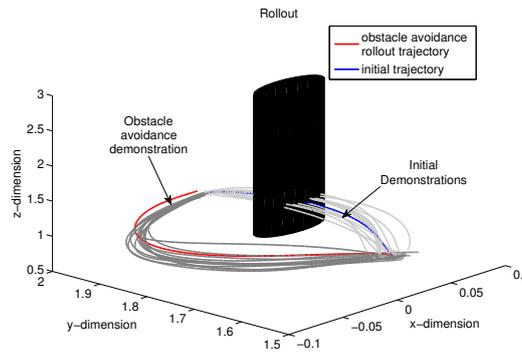
In our experiments, we tried to constraint the recombination weights of Equation 5.1b from reaching high values to keep the coupling term bounded at all times. If the weights are allowed to grow too large, the DMP can deviate from its path due to some unforeseen input and this can create a feedback loop that keeps diverging the DMP.

The resultant obstacle-avoidance behaviour of our unrolled coupling term is comparable to humans, as shown in Figure 7.8. When the obstacle is the same size and at the same position as during the recording of the human demonstrations, the resultant trajectory falls well within the variance of the demonstrations of the

subject. This shows that our approach is able to avoid obstacles in a human-like fashion. Even when we changed sizes and positions of the obstacle, as well as



(a)



(b)

Figure 7.8: Our obstacle avoidance formulation rolled out for a mean initial demonstration produces a likely obstacles avoidance trajectory both in case of (a) sphere and (b) cylinder. The grey trajectories show initial demonstrations by subjects. Blue trajectory was the initial DMP without an obstacle which results in the red trajectory in the presence of obstacles. Our unrolled trajectory looks a lot behaviourally like a human demonstration of obstacle avoidance.

duration of the movement, the coupling term works well, showing robustness to slight changes in the environment. Figure 7.9 shows the behaviour of the coupling term for a range of movement durations for the obstacle avoidance movement.

However, there can be situations, for example with an extremely big obstacle, when the coupling term does not manage to avoid the obstacle any more. Also, majority of our experiments are conducted on simple obstacles in static environments. This raises concerns over the effectiveness of our method for moving obstacles,

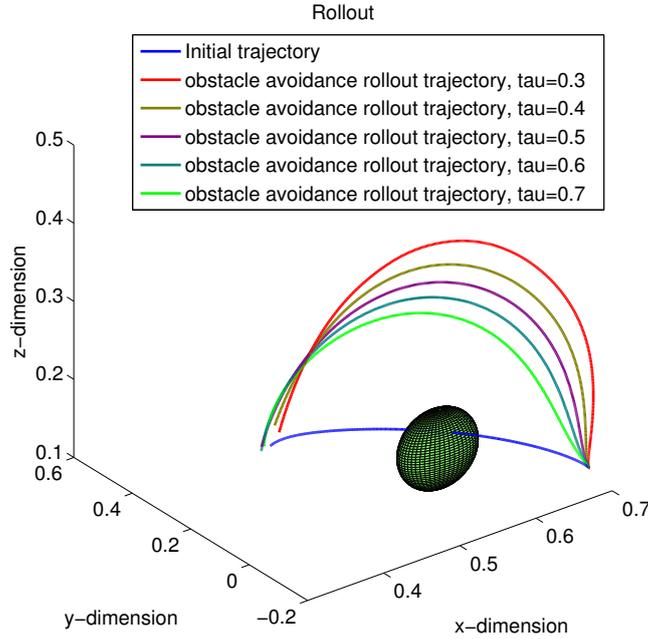


Figure 7.9: Different obstacle avoidance trajectories for different movement duration. For faster trajectories, we observe a larger deviation due to a higher velocity of movement. The coupling term is able to avoid the obstacle well, showing robustness to duration.

or complex non-convex obstacles. In such situations, when our method fails, we might require replanning the original movement, or using reinforcement learning to fine-tune specific components of the coupling term.

7.5 Automatic Relevance Determination on basis of features

Automatic Relevance Determination contributes a lot to the robustness of our coupling term in modelling human demonstrations. Because of higher flexibility, the ARD model fits the target coupling term better than usual linear regression, as well as brings down the dimension of our basis to about 50 from 615 in some dimensions. However, this comes at a cost. Since, we do impose an external constraint on our weights in our approach of ARD (Chapter 5) our weights grow very large.

Some statistics on the effect of ARD on our data are shown in Table 7.1.

Thus, even though it validates our hypothesis that the task of avoiding obstacles

| | | Dimension x | Dimension y | Dimension z |
|--|--------|-------------|-------------|-------------|
| <i>Mean for Regression Per Subject</i> | | | | |
| w_min | ARD | -5.34E5 | -3.05E5 | -1.70E5 |
| | No ARD | -15.10 | -38.26 | -24.92 |
| w_max | ARD | 4.73E5 | 8.47E5 | 6.79E5 |
| | No ARD | 17.09 | 40.32 | 26.79 |
| Mean sq. | ARD | 0.50 | 5.29 | 1.47 |
| Error | No ARD | 1.48 | 13.54 | 3.89 |
| Basis | ARD | 48.89 | 62.67 | 40.28 |
| Functions | No ARD | 615 | 615 | 615 |
| <i>Regression Across subjects</i> | | | | |
| w_min | ARD | -7.9E5 | -1.5E6 | -1.88E6 |
| | No ARD | -144.01 | -685.25 | -208.46 |
| w_max | ARD | 2.7E5 | 2.10E6 | 1.90E6 |
| | No ARD | 145.09 | 1224.35 | 710.36 |
| Mean sq. | ARD | 83 | 189 | 175 |
| Error | No ARD | 615 | 615 | 615 |
| Basis | ARD | 6.22 | 17.40 | 7.37 |
| Functions | No ARD | 10.03 | 68.24 | |

Table 7.1: Table showing the results of regression with and without ARD. Though ARD brings down the dimension of the basis function by a lot and decreases the mean squared error, the learnt weight magnitudes shoot up. This can lead to divergence of the DMP, and hence is not suggested.

can be done with a sparser basis, we cannot use these weights for unrolling our DMP. A DMP with very large magnitudes of coupling term weights often diverges due to the feedback effect explained before. However, if we could constraint the weights of our model, by using a forced regularizer, or added external constraints, we might be able to use this much sparser set of basis vectors.

7.6 Experiments on a simulated robot arm

To demonstrate the feasibility of our work on a realistic robotic set-up, we implemented the DMP system with obstacle avoidance on a simulated robot arm,

consisting of a bi-manual set-up of two KUKA Lightweight arms and an active vision head. Point-to-point movements with obstacle avoidance were executed with one arm. The DMP system created desired trajectories in the 3D task space of the robot, while an inverse kinematics formulation with null space optimization, followed by an inverse dynamics controller, realized these desired trajectories. This is a direct application of our method described above.

Some snapshots of the movement can be seen in Figure 7.10, and in Figures A.1, A.2, A.3 in the Appendix.

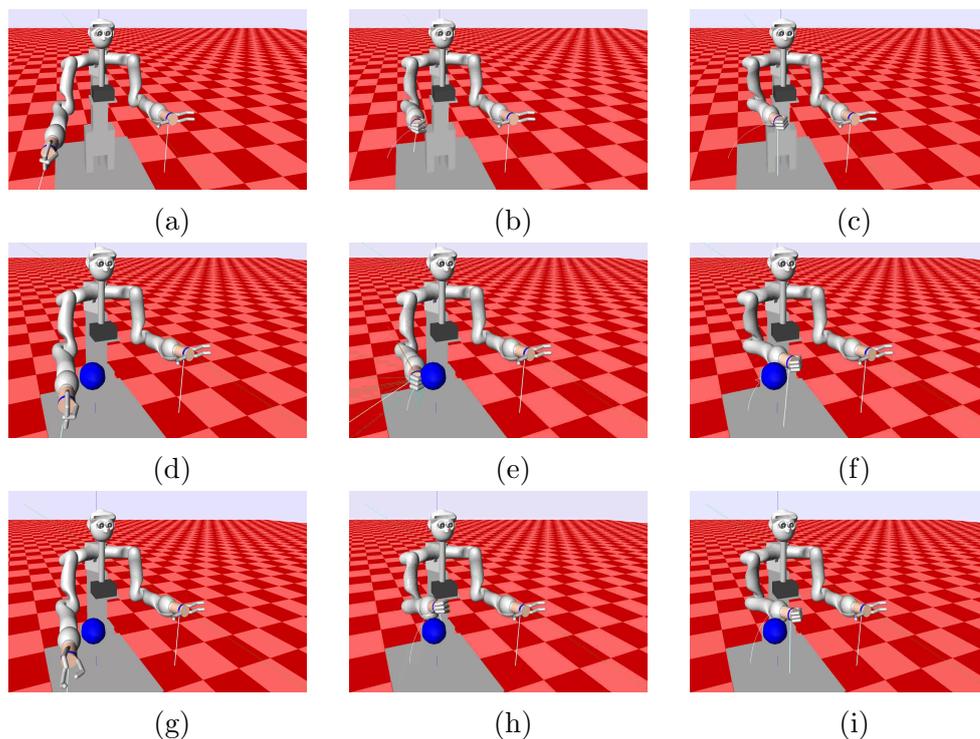


Figure 7.10: **Simulation Experiment 1:** (a), (b), (c) show the initial trajectory. (d), (e), (f) show the initial trajectory without any obstacle avoidance in the presence of a spherical obstacle. The end-effector hits the obstacle. (g), (h), (i) show the obstacle avoidance trajectory obtained using the coupling term. The end-effector manages to avoid the obstacle by moving over it.

CHAPTER 8

FUTURE WORK

Our experiments on coupling terms for obstacle avoidance, as we have described so far, point out some flaws, which can be addressed in future work.

1. Our coupling term is purely local and hence is a sub-optimal solution. It might also fail to find a solution to reach the goal, and get stuck, even when a solution exists. In such a scenario, we propose planning a new original path and starting all over again or using Reinforcement Learning to fine tune specific components of the coupling term.
2. Our coupling term does not take the speed of the obstacle into account. This feature might be important when avoiding moving obstacles in a dynamic environment. One simple work around is using the relative velocity between the end-effector and obstacle in Equation 4.1. But it might be useful to add more informative features about obstacle velocity.
3. Currently, we use only the end-effector center as the point that is avoiding obstacles. For a more robust obstacle avoidance with a real robot, it might be useful to include more points on the manipulator arm and robot body, like the elbow in our obstacle avoidance formulation.
4. The coupling term is unable to model the target C_t across different obstacle avoidance behaviours. We might need an additional hidden variable that chooses the behaviour needed to be modelled by the coupling term.

Apart from the coupling terms for obstacle avoidance, learning coupling terms can be applied to almost any task that needs an online modification of a pre-planned

path. For example, avoiding robot's own body and joint angle limits. These can be directly extrapolated from our current work. Bi-manual tasks that need force feedback for coordination, or learning sensor models that excite a particular behaviour can be other applications. By comparing with learnt sensor models from previous experiences, coupling terms can be associated with skill memories. In our work, we hand-design our features to some extent. It would be interesting to be able to learn these features automatically from data.

Online modification of a plan based on feedback is an important step towards autonomous manipulation. We try to introduce the idea of learning these coupling terms either from human demonstrations or simulated experiments in application to obstacle avoidance. More work needs to be done on finding ways that can learn powerful coupling terms for other tasks.

CHAPTER 9

CONCLUSION

Interpreting the environment and modifying a planned strategy based on the feedback can be implemented using coupling terms in DMPs. Here, we have used coupling terms to equip the DMP with obstacle avoidance features that remain dormant most of the time, except when an obstacle is detected in proximity. In this work, we propose a method to learn this coupling term from human demonstrations using a basis of simple features. We have shown the potential of our coupling term function to model different kinds of obstacle avoidance behaviours. Furthermore, we have shown that the open parameters of the proposed coupling term formulation can be learnt from human demonstrations. Our evaluation demonstrates that our formulation is capable of reproducing human obstacle-avoidance behaviour in a reactive manner quite well. Also, when unrolled on a new obstacle, the coupling term does fairly well over a range of sizes and positions of obstacles, as well as movement durations.

This work forms a good foundation for learning other coupling terms that use feedback from the environment to modulate a planned trajectory online. Several situations like joint angle limits, a robot avoiding its own body, etc can be directly extrapolated from and combined with our work on obstacle avoidance.

APPENDIX

A.1 Experiments on a simulated robot arm

To demonstrate the feasibility of our work on a realistic robotic set-up, we implemented the DMP system with obstacle avoidance on a simulated robot arm, consisting of a bi-manual set-up of two KUKA Lightweight arms and an active vision head. Point-to-point movements with obstacle avoidance were executed with one arm. The DMP system created desired trajectories in the 3D task space of the robot, while an inverse kinematics formulation with null space optimization, followed by an inverse dynamics controller, realized these desired trajectories. This is a direct application of our method described above.

Some snapshots of the movement can be seen in Figures 7.10, A.1, A.2, A.3

As can be seen, our approach realizes successful obstacle avoidance behaviour in a natural looking way. The results can be found at <http://www-clmc.usc.edu/~arai/pub/humanoids2014.mov>.

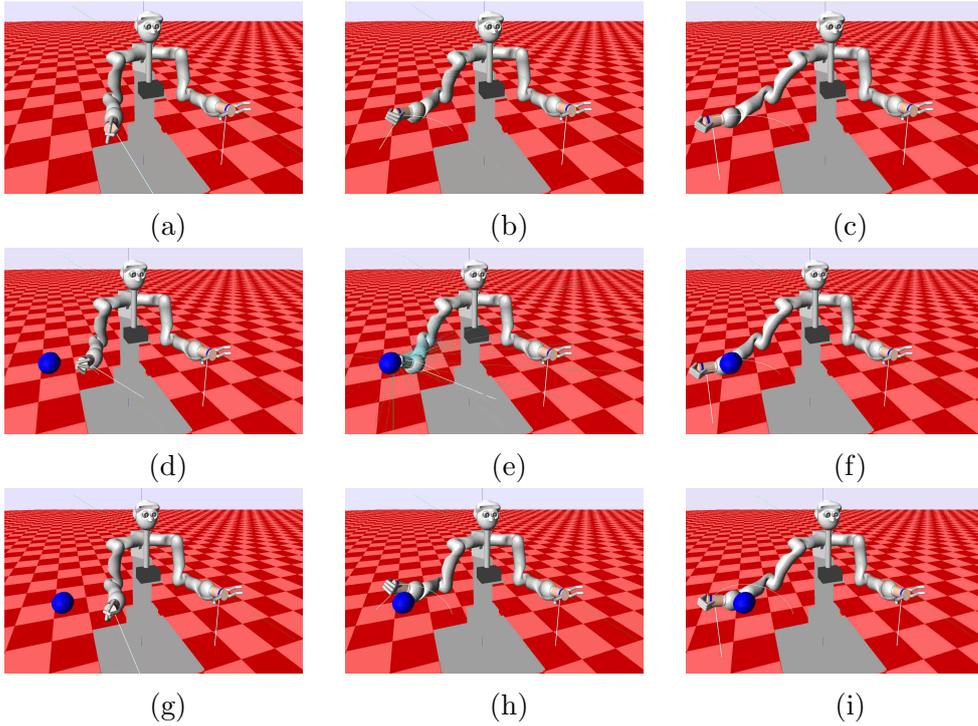


Figure A.1: **Simulation Experiment 2:** This trajectory is faster than simulation experiment 1. (a), (b), (c) show the initial trajectory. (d), (e), (f) show the initial trajectory without any obstacle avoidance in the presence of a spherical obstacle. The end-effector hits the obstacle. (g), (h), (i) show the obstacle avoidance trajectory obtained using the coupling term. The end-effector manages to avoid the obstacle by moving over it.

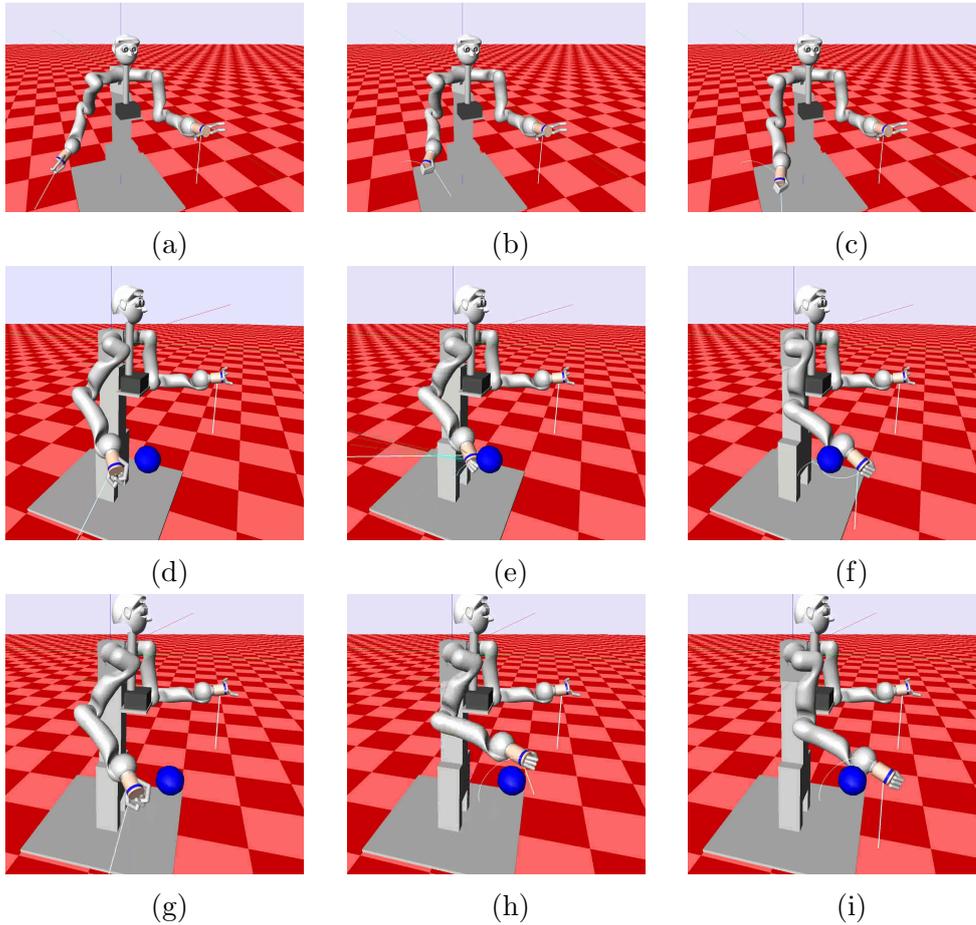


Figure A.2: **Simulation Experiment 3:** This trajectory is slower than experiments 1 and 2. (a), (b), (c) show the initial trajectory. (d), (e), (f) show the initial trajectory without any obstacle avoidance in the presence of a spherical obstacle. The end-effector hits the obstacle. (g), (h), (i) show the obstacle avoidance trajectory obtained using the coupling term. The end-effector manages to avoid the obstacle by moving over it.

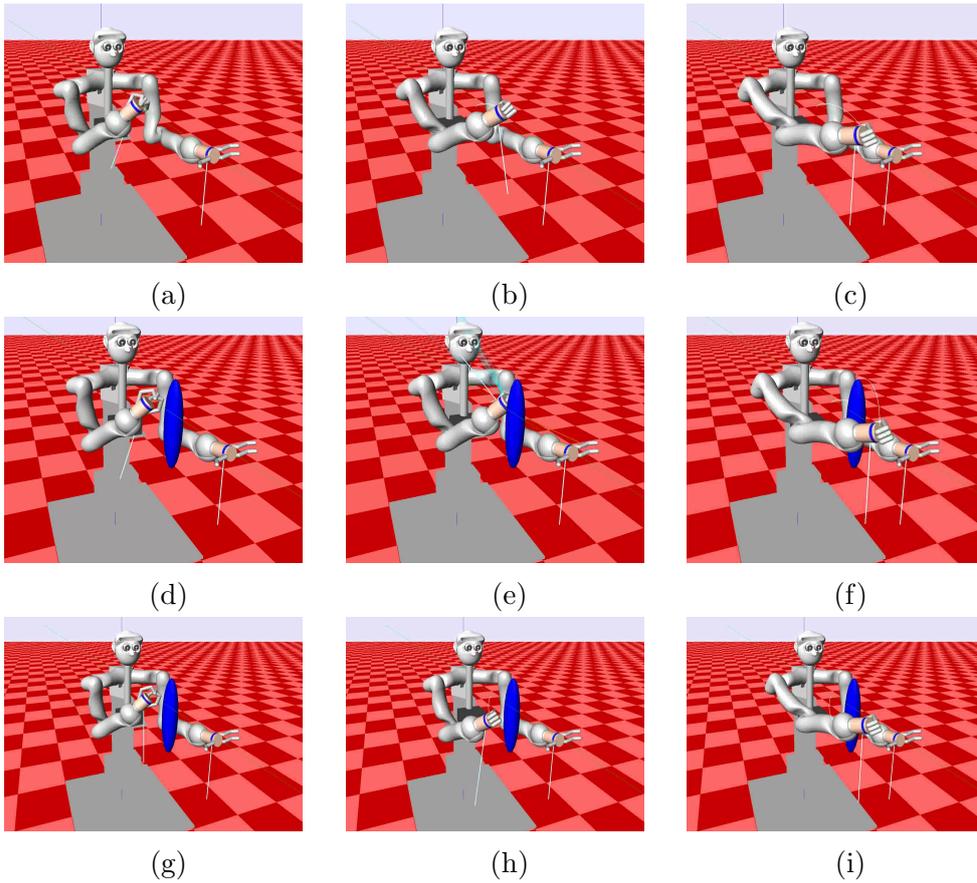


Figure A.3: **Simulation Experiment 4:** (a), (b), (c) show the initial trajectory. (d), (e), (f) show the initial trajectory without any obstacle avoidance in the presence of a ellipsoidal obstacle. The end-effector hits the obstacle. (g), (h), (i) show the obstacle avoidance trajectory obtained using the coupling term. The end-effector manages to avoid the obstacle by moving around it.

BIBLIOGRAPHY

- [1] Michael Barbehenn, Pang C Chen, and Seth Hutchinson. An efficient hybrid planner in changing environments. In *Robotics and Automation, 1994. Proceedings., 1994 IEEE International Conference on*, pages 2755–2760. IEEE, 1994.
- [2] Johann Borenstein and Yoram Koren. The vector field histogram-fast obstacle avoidance for mobile robots. *Robotics and Automation, IEEE Transactions on*, 7(3):278–288, 1991.
- [3] Rosen Diankov and James Kuffner. Randomized statistical path planning. In *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, pages 1–6. IEEE, 2007.
- [4] Andrej Gams, Auke J Ijspeert, Stefan Schaal, and Jadran Lenarčič. On-line learning and modulation of periodic movements with nonlinear dynamical systems. *Autonomous robots*, 27(1):3–23, 2009.
- [5] Andrej Gams, Bojan Nemec, Auke Jan Ijspeert, and Aleš Ude. Coupling movement primitives: Interaction with the environment and bimanual tasks.
- [6] Auke Jan Ijspeert, Jun Nakanishi, and Stefan Schaal. Learning attractor landscapes for learning motor primitives. Technical report, 2002.
- [7] Ioannis Iossifidis and G Schoner. Autonomous reaching and obstacle avoidance with the anthropomorphic arm of a robotic assistant using the attractor dynamics approach. In *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, volume 5, pages 4295–4300. IEEE, 2004.

- [8] Mrinal Kalakrishnan, Sachin Chitta, Evangelos Theodorou, Peter Pastor, and Stefan Schaal. Stomp: Stochastic trajectory optimization for motion planning. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 4569–4574. IEEE, 2011.
- [9] Oussama Khatib. Real-time obstacle avoidance for manipulators and mobile robots. *The international journal of robotics research*, 5(1):90–98, 1986.
- [10] Tomas Lozano-Perez. Spatial planning: A configuration space approach. *Computers, IEEE Transactions on*, 100(2):108–120, 1983.
- [11] Hans P Moravec. Sensor fusion in certainty grids for mobile robots. *AI magazine*, 9(2):61, 1988.
- [12] Randal C. Nelson and Jhon Aloimonos. Obstacle avoidance using flow field divergence. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 11(10):1102–1106, 1989.
- [13] Dae-Hyung Park, Heiko Hoffmann, Peter Pastor, and Stefan 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.
- [14] Peter Pastor, Heiko Hoffmann, Tamim Asfour, and Stefan 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.
- [15] Peter Pastor, Mrinal Kalakrishnan, Franziska Meier, Freek Stulp, Jonas Buchli, Evangelos Theodorou, and Stefan Schaal. From dynamic movement primitives to associative skill memories. *Robotics and Autonomous Systems*, 61(4):351–361, 2013.
- [16] Peter Pastor, L. Righetti, Mrinal Kalakrishnan, and S. Schaal. Online movement adaptation based on previous sensor experiences. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 365–371, Sept 2011.

- [17] Sean Quinlan and Oussama Khatib. Elastic bands: Connecting path planning and control. In *Robotics and Automation, 1993. Proceedings., 1993 IEEE International Conference on*, pages 802–807. IEEE, 1993.
- [18] Nathan Ratliff, Matthew Zucker, J Andrew Bagnell, and Siddhartha Srinivasa. Chomp: Gradient optimization techniques for efficient motion planning. In *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, pages 489–494. IEEE, 2009.
- [19] Reid Simmons. The curvature-velocity method for local obstacle avoidance. In *Robotics and Automation, 1996. Proceedings., 1996 IEEE International Conference on*, volume 4, pages 3375–3382. IEEE, 1996.
- [20] Michael E Tipping. Sparse Bayesian learning and the relevance vector machine. *The Journal of Machine Learning Research*, 1:211–244, 2001.
- [21] W Warren, B Fajen, and D Belcher. Behavioral dynamics of steering, obstacle avoidance, and route selection. *Journal of Vision*, 1(3):184–184, 2001.
- [22] Eiichi Yoshida and Fumio Kanehiro. Reactive robot motion using path replanning and deformation. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 5456–5462. IEEE, 2011.