A Dynamical Systems Approach to Adaptive Sequencing of Movement Primitives

Tobias Luksch[1,2], Michael Gienger[1], Manuel Mühlig[1], Takahide Yoshiike[1]

[1] Honda Research Institute Europe, Offenbach, Germany

[2] Robotics Research Lab, University of Kaiserslautern, Germany

Abstract

This paper introduces a control concept for motion generation of redundant robots based on combinations of movement primitives (MP). It addresses the question of how to create continuous and smooth sequences of actions or transitions between different motion skills while avoiding the necessity of recurrent planning. MPs are defined on task coordinates and modeled as dynamical systems with attractor behavior featuring additional signals to ease their coordination. Sequences and transitions between skills are realized in a unified way as bifurcating dynamical systems based on continuous-time recurrent neural networks. The neural output is used as activation signal for MPs. It is shown how the parameters of these dynamical systems can be chosen to generate a desired behavior. First results are shown in a physical simulation environment on a high-DoF robot with human-like upper body. The system can create smooth transients of MPs in sequences as well as during changes of strategies, notably showing more than only local adaptation capabilities.

Keywords: adaptive control, movement primitives, dynamical systems

1 Introduction

Recent biological and neuroscientific studies suggest that complex movement skills found in animals and humans are combinations of motor primitives [1–3]. The notion of such primitives spans from limb controlling attractor dynamics located in the spinal cord to behavioral relevant motions coded in the motor cortex. A growing body of literature is dedicated to analyzing how sequencing and the selection of sequences might be represented by the nervous system [4,5]. Results imply subconscious control of short sequences, hierarchical organization, concurrent executing on different levels, as well as continuous blending from the selection of one action sequence to another.

In line with these findings, the present work suggests a similar control scheme for robot motion skills. Non-primitive skills are to be composed by blending and sequencing of primitive motions, called movement primitives (MPs) in the following. The goal of the presented work is to improve the state of the art by allowing: smooth transients between MPs; adaptive sequences of MPs including concurrent execution; automatic and continuous transient of action sequences without the necessity of repeated planning; recovery from disturbances and errors on all levels of the control system, going beyond only local adaptation. Both low-level MPs and higher level, coordinating control units responsible for sequencing and arbitration will be represented by dynamical systems.

The following chapter will briefly introduce the concept of a movement primitive and how it is modeled in this work. Chapter 3 presents the main focus of this paper, namely how MPs can be coordinated to generate desired skills. It is shown how the suggested neural dynamics can encode sequences as well as continuous switching of skills based on perceptual feedback. The experimental results of Chapter 4 will illustrate the functional principles of the proposed methods. The paper closes with summary and outlook.

2 Movement Primitives as Dynamical Systems

A movement primitive as it is understood in this paper is a unit generating control commands in task or joint coordinates to implement an elementary action, e. g. a reaching motion towards an object. One MP does not necessarily affect all degrees of freedom of the robot. This work utilizes dynamical systems (DS) defined on the MP's state space as generation methods for control commands:

$$\dot{\boldsymbol{y}} = f(\boldsymbol{y}, \boldsymbol{p}_{\mathrm{DS}})$$

where $\boldsymbol{y} = s(\boldsymbol{x})$ is a subset of the full state $\boldsymbol{x}$ relevant for the MP, and $\boldsymbol{p}_{\mathrm{DS}}$ is a parameter vector describing the behavior of the DS. There are various benefits of this approach: due to the inherent state feedback, DSs are robust against disturbances or inaccuracy of sensors; most models are compact and have low computational cost for progression; DSs show good generalization capabilities; depending on the model used, DSs are time invariant.

There is an increasing number of examples of using DSs for motion representation to be found in literature. One of the most prominent approaches are the Dynamic Movement Primitives (DMP) [6]. Based on Gaussian blending of simple attractor dynamics, they use an additional canonical system to represent progression. Other approaches include the usage of GMM [7], SEDS [8], HMM [9], or RNN [10,11]. It is not the intention of this work to find a novel DS representation for MPs. Any of the models mentioned above can be used. Rather, in addition to the motion generating DS, the interface of a MP is extended: Beside the current state, each MP receives an activation signal $a \in [0, 1]$ that is used to scale the outgoing control command. Furthermore, a prediction $\hat{\boldsymbol{y}}$ of the MP's state vector will be generated, using the current state and control command. It is used to calculate a prediction error p reflecting the difference of what is observed and what the MP is expecting to happen. Finally, each MP generates a goal distance $g \in [0, 1]$ and a responsibility value $r \in [0, 1]$ based on the current state. These MP signals p, g, and r can be interpreted as pre-processed (from the MP's point of view) perceptual information and are used to influence higher level activation dynamics as shown in the next chapter.

The described MPs work independently of the underlying robot controller (position-based, torque-based, …). For the experiments of this work, the kinematic task space control commands generated by the MPs are mapped to joint velocities using a resolved motion rate controller including null space optimization. Which task coordinates have to be considered depends on the output of those MPs that are currently activated.

The choice of task coordinates for a given MP can heavily influence its generalization capabilities and re-usability. For instance, while grasping an object the end-effector pose can be controlled relative to the object's frame instead of using global or robot-local coordinates. As a result, the motion is independent of the object pose and the robot's kinematic layout. Disturbances to the end-effector or object pose can be compensated directly by the DS. By defining the DSs on such relative coordinates, no additional parameters have to be passed to MPs. The activation a remains the only input. Further details on choosing suitable frames of reference and their biological plausibility can be found in [12].

As an example, consider a MP executing a reaching movement towards an object as preparation for grasping. Its DS controls the end-effector within the coordinate frame of the object. The goal distance g can be set to the distance of hand and object, normalized by the reaching range of the robot. Based on the object position in robot-local coordinates, the responsibility r can capture the working space of the robot, e. g. by approximating it using a GMM. The MP would then express responsibility only if the object is located within reach. Finally, the MP would assume the object to remain stationary by setting the predicted object-relative end-effector pose to $\hat{\boldsymbol{y}} = \boldsymbol{y} + \Delta t \cdot \dot{\boldsymbol{y}}$, where Δt is the step length of time-discrete control loop. This prediction results in large disturbances of the object or end-effector pose leading to a large prediction error p.

3 Coordination of Movement Primitives

Following the findings in neuroscience as stated above, motion skills of increasing complexity shall be achieved by blending and sequencing of MPs. This is done by generating a continuous stream of activation signals $\boldsymbol{a}$. Such a stream can represent a sequence by parallel and serial blending of MPs. But it should be possible to influence the activation stream, e. g. to recover from errors or to smoothly blend to another sequence of actions.

In this paper it is assumed that all MPs and skills are either designed or already learned by imitation or experience. It is not the intention to create novel combinations of MPs by e. g. logical planning. But the control system should be able to blend between already known MPs and skills based on sensory feedback. The skill representation itself should allow adaptation to changes in the environment. The necessity of having to re-plan constantly should be avoided, as planning tends to be slow, neglects experience from similar, previous situations, and is not biologically plausible on the subconscious level of behavior generation.

One possible way to represent skills and strategies are finite state machines (FSM). While being easy to design and understand, FSMs have several drawbacks, such as the need to explicitly define every transition, discrete switches between states, or dependencies on thresholds. Also, it can be difficult to implement learning or continuous, non-discrete adaptation and optimization and to synchronize concurrent flow of transitions.

Instead of FSMs, this work suggests to use a DS approach to generate continuous flows of MP activations and thus representing motion skills. Constant feedback of sensor information results in bifurcation of these dynamics, thus generating smooth transients between MPs. This mechanism can uniformly create sequences as well as transitions of strategies, e. g. for error recovery. Furthermore, such an "activation dynamics" is wrapped in the same interface as the MPs described in Chapter 2, i. e. having an activation input and calculating signals p, g, and r. In this way, high level skills can recruit lower level skills as well as MPs, creating a hierarchical, homogeneous control system.

Some approaches using a DS representation to activate control units can be found in literature. For instance, the behavior-based control community adapted this idea using specialized competitive dynamics to arbitrate behaviors [13,14]. Other work encodes MPs as well as their coordination as single recurrent neural networks [10]. While this method allows for simultaneous learning of MPs, sequences, and expected perceptual feedback, it is not sure how well such monolithic systems will scale. Also, it is difficult to interpret the inner workings of the network.

A few approaches use neural fields [15] as modeling tool for activations. While they are suited best to stabilize values of a continuous metric as in low-level direction commands or cognitive tasks, activated pools of neurons at a certain position in the field can also be used to represent the activity of a discrete MP [16,17]. Recently, there have been attempts to encode action sequences as DSs based on neural fields [18]. While this approach is promising, the suggested system using a simple ordinal dynamics for representing serial order lacks some of the desired capabilities, e. g. error recovery by strategy transitions.

3.1 Activation Dynamics

While the representation of activation dynamics used in this work was inspired by neural field formulations as cited above, it neglects the field characteristics. As the activation of individual MPs is to be determined, a continuous field with fixed lateral influence is not required. Instead, a neural dynamics as in continuous-time recurrent neural networks (RNN) is used as it can model similar beneficial characteristics, like the possibility of conditioned selection by pre-activation based on sensor data, or inherent hysteresis but still fast behavior in case of large stimuli [19].
Each neuron of the RNN represents one of the MPs controlled by the activation dynamics. The dynamics of the potential u_i of neuron i follows the equation

$$\tau \dot{u}_i = -u_i + h + w_0 \cdot f_\sigma(u_i) + s_i \tag{1}$$

with time constant τ, resting level $h < 0$, self-excitation weight w_0, and additional stimulus or inhibition s_i (see below). The output of a neuron $f_\sigma(u_i)$ is used as activation value for the corresponding MP. As output function, the logistic function is used, with β defining the steepness of the sigmoid:

$$f_{\text{sigmoid}}(u) = f_\sigma(u) = \frac{1}{1 + e^{-\beta u}} \tag{2}$$

The neural dynamics defined by Eq. 1 and 2 show a bistable behavior that is essential for its use as MPs activation dynamics. It allows neural activity that was initiated by time-dependent stimuli to become self-sustained in the absence of external input signals. Only large excitatory or inhibitory input will destabilize one of the fixed points and trigger a transient between resting level and activation level. Thus, the neuron also exhibits filtering and hysteresis properties.

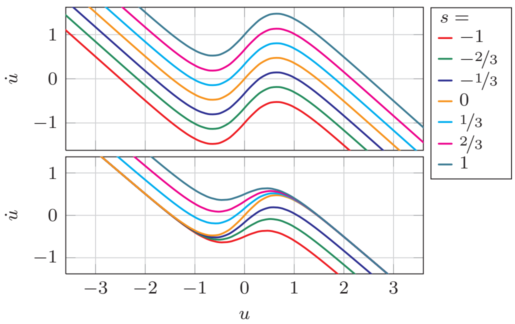

Figure 1: Phase plot of neuron potential for various stimuli. Top: Eq. 1, bottom: Eq. 3 ($h = -1.5$, $w_0 = 3$, $\beta = 3$).

In case of no external stimulus, the two stable fixed points occur at the resting level h and at $h+w_0$, with the weight of self-excitation normally chosen as $w_0 = -2 \cdot h$ to achieve symmetry of fixed points. **Figure 1** (top) illustrates the change of neuron potential given different external stimuli. The zero-crossings with negative gradient of each line indicate stable fixed points.
External input s_i shifts the fixed points of the neural dynamics. Very high or low values enlarge their distance and thus can increase transition time when the stimulus changes. To avoid this, the Eq. 1 can be adapted to

$$\begin{aligned}\tau \dot{u}_i = {} & -u_i + h + w_0 \cdot f_\sigma(u_i) \\ & + \begin{cases} (1 - f_\sigma(u_i)) \cdot s_i, & s_i \geq 0 \\ f_\sigma(u_i) \cdot s_i, & s_i < 0 \end{cases}\end{aligned} \tag{3}$$

This results in qualitatively similar behavior, but confines the location of the stable fixed points to the interval $[h, h + w_0]$, as illustrated in **Figure 1** (bottom). It is a drawback of this variant that the scope for pre-activation (or pre-inhibition) of a neuron is reduced. Both Eq. 1 and Eq. 3 were used for experiments in this work.
The external stimulus s_i of Eq. 1 and Eq. 3 includes the recurrent influence from other neurons and amounts to

$$\begin{aligned} s_i = {} & a_{\text{Skill}} && \text{(base excitation)} && (4)\\ & - c_0 \cdot \sum_{j \in \mathcal{R}_i} g_j && \text{(goal distance term)} && (5)\\ & - c_1 \cdot \sum_{k \in \mathcal{I}_i} f_\sigma(u_k) && \text{(inhibition term)} && (6)\\ & - c_2 \cdot (1 - r_i) && \text{(responsibility)} && (7)\\ & - c_3 \cdot p_i && \text{(prediction error)} && (8) \end{aligned}$$

Further terms consider the corresponding MP's responsibility r_i, prediction error p_i, and requirements (in terms of other MP's goal distances g_j). In addition, the activation $a_{\text{Skill}} \in [0, 1]$ of the controlling, high-level skill containing the activation dynamics is added as base excitation. $\mathcal{R}_i$ describes a set of indices of MPs that are preceding MP i, and $\mathcal{I}_i$ a set of indices of MPs inhibiting MP i.
Considering the knowledge about the bifurcation behavior of Eq. 1 or 3, it is not difficult to find suitable parameters $c_{[\cdot]}$ to generate the desired behavior. For instance, given $h = -1.5$, $w_0 = 3$, and $\beta = 3$, the upper (lower) fixed point is destabilized for inputs $s_i < -0.48$ ($s_i > 0.48$). With a full base excitation of $a_{\text{Skill}} = 1$, setting parameter $c_2 = 1$ would result in the neuron to transition into the activation level for a responsibility of $r_i > 0.5$, and it would sustain that state even if the responsibility drops to Zero. Similar considerations help to find the other parameters.

3.2 Encoding Sequences and Arbitration

Using the lateral influence and additional stimuli, sequences and transitions between sequences can be achieved. In sequences, connection weights inhibit a neuron's activation if subsequent neurons are activated. Selection between options is accomplished by mutual inhibition of neurons representing different skills or sequences. Selecting these inhibition schemes is done by including the appropriate indices of inhibiting MPs or activation dynamics in $\mathcal{I}_i$ (Eq. 6). If the currently active skill is inhibited by e. g. a large prediction error (Eq. 8), the next best skill will be able to get active. "Next best" in this case corresponds to the neuron with the highest pre-activation, i. e. with lowest prediction error or highest responsibility value over the preceding timesteps.

Beside the inhibition term, encoding of sequences makes use of the goal distance values. A MP can be inhibited as long as preceding MPs are not close to their goal (Eq. 5). The goal distance is not necessary an Euclidean distance, but can be an arbitrary distance measure as the state of a MP could for instance also include force values.

4 Experimental Results

First results of the presented control concept are based on reach-and-grasp experiments within a sophisticated physical simulation environment. The utilized robot is equipped with a human-like upper body, two three-finger hands, and features a total 38 degrees of freedom. It is placed before a table and has to reach for a ball which can be disturbed by placing it to arbitrary position or by setting a horizontal velocity (**Figure 2**). The robot is assumed to be able to perceive the location of the ball and can measure various touch, force, and torque information.

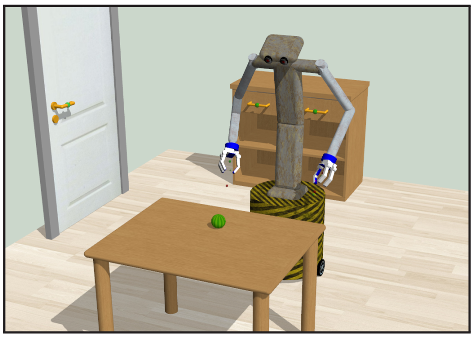

Figure 2: Screenshot of the simulation environment

The control hierarchy used for the experiments is shown in **Figure 3**. All low-level MPs (gray boxes) use linear attractor dynamics on various task coordinates, e. g. relative hand-object transformation (X LeftPG-GB0). The high-level control units (red boxes) make use of the activation dynamics as described in the previous chapter. Green arrows indicate activation signals, blue arrows depict task commands.

The implementation for the experiments below uses Eq. 3 for updating the neuron potentials. The parameters used are resting level $h = -1.5$, self-excitation $w_0 = -2 \cdot h = 3$, sigmoid function steepness $\beta = 3$, and time constant $\tau = 0.1$. The factors governing the stimulus term are set to $c_0 = 0.5$, $c_1 = 4$, $c_2 = 2$, and $c_3 = 4$ for all involved activation dynamics.

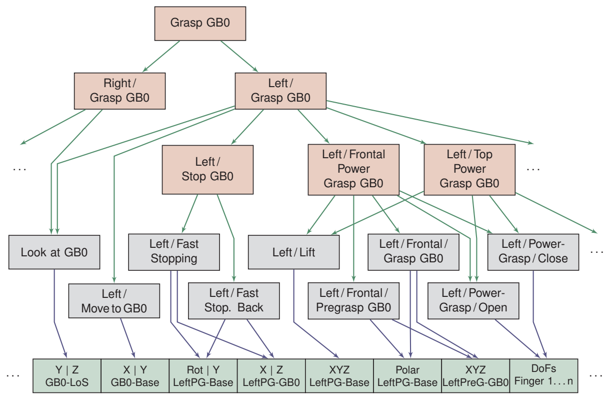

Figure 3: Control hierarchy (green box: task coordinates, gray box: low-level MPs, red box: activation dynamics)

4.1 Undisturbed Grasping Sequence

A closer look at the grasping skill Left/Frontal Power Grasp GB0 shall illustrate the inhibition and requirement sets $\mathcal{R}_i$ and $\mathcal{I}_i$. This activation dynamics controls five low-level MPs (right-most gray boxes of **Figure 3**) and organizes them as a flexible sequence to achieve object grasping. **Table 1** lists the MPs as well as the corresponding inhibition and requirement sets. An "x" denotes that the index of that row's MP is an element of the set. For instance, $\mathcal{R}_1$ contains the indices 0 and 2, denoting that the movement to the final grasping pose should only be executed if the goal distance of Frontal Pre-Grasp and Power-Grasp Open is small, i. e. if the pre-grasping pose is reached and the robot's hand is opened. Similarly, index 3 within $\mathcal{I}_2$ indicates that motion of opening the robot's hand is inhibited while closing it.

Name of MP	$\mathcal{R}_0$	$\mathcal{R}_1$	$\mathcal{R}_2$	$\mathcal{R}_3$	$\mathcal{R}_4$	$\mathcal{I}_0$	$\mathcal{I}_1$	$\mathcal{I}_2$	$\mathcal{I}_3$	$\mathcal{I}_4$
0: Frontal Pre-Grasp	-	x	-	-	-	-	-	-	-	-
1: Frontal Grasp	-	-	-	x	x	x	-	-	-	-
2: Power-Gr. Open	-	x	-	-	-	-	-	-	-	-
3: Power-Gr. Close	-	-	-	-	x	-	-	x	-	-
4: Lift	-	-	-	-	-	x	x	-	-	-

Table 1: Inhibition and requirement sets for Frontal Power Grasp GB0 (the "Left/" prefix has been skipped).

Figure 4 visualizes the mechanisms of the activation dynamics during a grasping sequence without disturbances. The top-most plot depicts the vertical position of the ball that is to be grasped. It is lifted from the table top starting at $t \approx 5.1$ sec by the Lift MP. The remaining four plots show (from top to bottom) the goal distances, stimuli, neuron potentials, and output values of the five controlled MPs or their corresponding neurons within the RNN. The neural output values are directly used as activation signals for the commanded MPs.

At $t \approx 0.8$ sec, the activation dynamics of the grasping skill is activated by higher levels of the control hierarchy, as can be seen by the increasing stimuli for all five neurons caused by the base excitation (Eq. 4). For both the Pre-Grasp and the Open MPs the stimulus is sufficiently high to destabilize the lower stable fixed point of the neural dynamics, resulting in a smooth transient to the higher fixed point and thus to an activation of the MPs. The stimulus of the Open MP is a bit lower because of its responsibility definition (Eq. 7) and causes a slower increase of the neuron's potential.

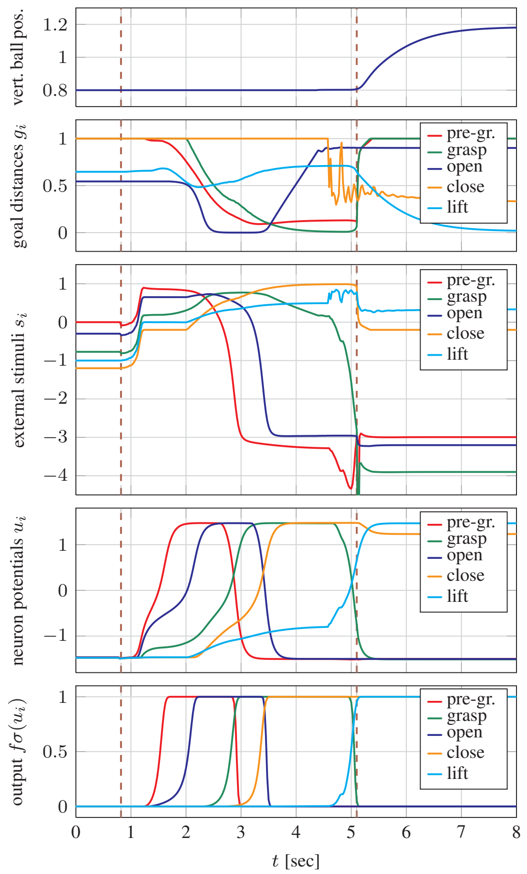

Figure 4: Inputs, neuron potentials, and output values of the grasping skill's RNN without external disturbances.

The decrease of the two MPs' goal distances also decreases the inhibition on the Grasp MP, allowing its potential to raise (Eq. 5, see also **Table 1**). This in turn causes an inhibition of the Pre-Grasp MP that is strong enough to lower its potential back to below resting level (Eq. 6). Such an interaction of excitation by decreasing goal distance of the preceding MP and inhibition by increasing activation of a subsequent MP is typically used for sequence encoding. Similar interplay is taking place for the succeeding parts of the sequence, namely the grasping of the object by closing the robot's hand, and lifting the object from the table. The Close MP's goal distance value exhibits some noise as the simulated measured grasping force is part of the goal distance calculation.

The activation dynamics shows the ability to generate smooth transients between MPs during progressing a sequence. As can be seen in the example above, also parallel activation of multiple MPs is supported. Local adaptation of disturbances is handled by the low-level MPs themselves, e. g. small shifts of the object during grasping would be compensated by the DSs of the Pre-Grasp and Grasp MPs. Further adaptation takes place within the sequence itself: If for instance the object slips from the robot's hand, the Close MP will indicate this with a high prediction error as it predicts a suitable grasping force when the hand is closed. The prediction error will self-inhibit the MP (Eq. 8), causing the sequence to re-enter at an earlier point resulting in a re-grasping behavior.

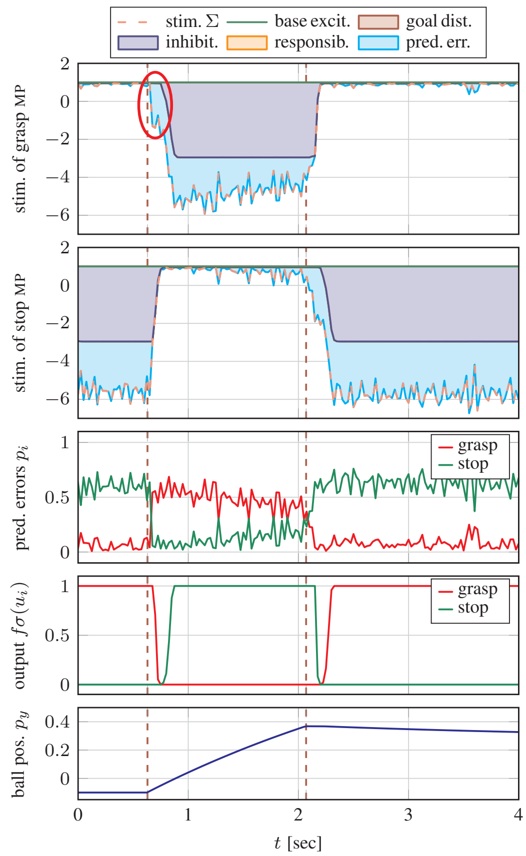

Figure 5: Transitions between skills for grasping a stationary ball and for stopping a rolling ball. The top two plots visualize the composition of the stimuli as in Eq. 4–8.

4.2 Transition of Skills due to Disturbance

This experiment illustrates the transition between skills caused by large external disturbances. It involves two skills, namely grasping a stationary ball (Grasp) and stopping a rolling ball by intercepting its trajectory with the robot's hand (Stop). As shown in **Figure 3**, these skills are controlled by a higher level competing activation dynamics with mutual inhibition of all neurons. **Figure 5** visualizes a situation where the robot is already reaching towards the ball lying on the table. At $t \approx 0.6$ sec, the ball is disturbed by adding a horizontal velocity as can be seen from the bottom-most plot showing the horizontal position of the ball. As the grasping skill predicts an approximately constant position of the ball, its prediction error increases. The prediction error values are noisy as artificial noise is added to the simulated sensor information to test robustness. The increase of prediction error results in self-inhibition (Eq. 8) of the corresponding neuron (red ellipse in stimulus plot), which again allows the ball stopping skill to be activated. The latter predicts a target moving at medium velocity, so its prediction error is low. As soon as the movement of the ball is stopped at $t \approx 2.1$ sec, the situation reverses and the ball can be grasped again.

5 Summary and Outlook

A control approach that continuously combines movement primitives to motion skills has been presented. Dynamical systems are used to model basic motion elements as well as sequences and decision making. Besides local adaptation, the suggested method is able to adapt to errors and disturbances on a more global level by smooth transients between sequences and skills.
Future work will include imitation learning of MPs and sequences as well as the analysis of scenarios including force feedback with application on physical robots.

References

[1] E. Bizzi, A. d'Avella, P. Saltiel, and M. Tresch, "Modular organization of spinal motor systems," *The Neuroscientist*, vol. 8, no. 5, pp. 437–442, 2002.

[2] T. Flash and B. Hochner, "Motor primitives in vertebrates and invertebrates," *Current Opinion in Neurobiology*, vol. 15, no. 6, pp. 660–666, 2005.

[3] M. Graziano, "The organization of behavioral repertoire in motor cortex," *Annu. Rev. Neurosci.*, vol. 29, pp. 105–134, 2006.

[4] S. Monsell, "Task switching," *Trends in cognitive sciences*, vol. 7, no. 3, pp. 134–140, 2003.

[5] B. Rhodes, D. Bullock, W. Verwey, B. Averbeck, and M. Page, "Learning and production of movement sequences: Behavioral, neurophysiological, and modeling perspectives," *Human movement science*, vol. 23, no. 5, pp. 699–746, 2004.

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

[7] M. Mühlig, M. Gienger, and J. Steil, "Interactive imitation learning of object movement skills," *Autonomous Robots*, vol. 32, pp. 97–114, 2012.

[8] S. Khansari-Zadeh and A. Billard, "Imitation learning of globally stable non-linear point-to-point robot motions using nonlinear programming," in *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2010, pp. 2676–2683.

[9] D. Kulić, W. Takano, and Y. Nakamura, "Towards lifelong learning and organization of whole body motion patterns," *Robotics Research*, pp. 87–97, 2011.

[10] J. Tani, "Learning to generate articulated behavior through the bottom-up and the top-down interaction processes," *Neural Networks*, vol. 16, no. 1, pp. 11–23, 2003.

[11] R. Reinhart and J. Steil, "Reaching movement generation with a recurrent neural network based on learning inverse kinematics for the humanoid robot icub," in *Proc. of IEEE/RAS International Conference on Humanoid Robots (Humanoids)*, 2009, pp. 323–330.

[12] M. Gienger, C. Goerick, and E. Körner, "Movement control in biologically plausible frames of reference," in *ISR / Robotik 2010*. VDE Verlag, 2010.

[13] H. Jaeger and T. Christaller, "Dual dynamics: Designing behavior systems for autonomous robots," *Artificial Life and Robotics*, vol. 2, no. 3, pp. 108–112, 1998.

[14] T. Bergener, C. Bruckhoff, P. Dahm, H. Janßen, F. Joublin, R. Menzner, A. Steinhage, and W. von Seelen, "Complex behavior by means of dynamical systems for an anthropomorphic robot," *Neural Networks*, vol. 12, no. 7, pp. 1087–1099, 1999.

[15] S. Amari, "Dynamics of pattern formation in lateral-inhibition type neural fields," *Biological Cybernetics*, vol. 27, no. 2, pp. 77–87, 1977.

[16] W. Erlhagen and G. Schöner, "Dynamic field theory of movement preparation." *Psychological review*, vol. 109, no. 3, p. 545, 2002.

[17] E. Bicho, W. Erlhagen, L. Louro, and E. Costa e Silva, "Neuro-cognitive mechanisms of decision making in joint action: a human-robot interaction study," *Human Movement Science*, 2011.

[18] Y. Sandamirskaya and G. Schöner, "Serial order in an acting system: a multidimensional dynamic neural fields implementation," in *Proc. of IEEE International Conference on Development and Learning (ICDL)*, 2010, pp. 251–256.

[19] R. Beer, "On the dynamics of small continuous-time recurrent neural networks," *Adaptive Behavior*, vol. 3, no. 4, pp. 469–509, 1995.