Adaptive Movement Sequences and Predictive Decisions based on Hierarchical Dynamical Systems

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

***Abstract*— This paper addresses the question of how to create adaptive and smooth sequences of actions and how to decide among skill options in a continuous manner without the necessity of recurrent planning. Motion generation is based on serial and parallel blending of movement primitives (MP). MPs are modeled as dynamical systems on task coordinates with attractor behavior and augmented with 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. Besides continuous feedback from the controlled MPs, the neural dynamics is influenced by a cost term from a future prediction to allow the inhibition of an action flow that is expected to fail.**

First results are shown in a physical simulation environment on a high-DoF robotic hand-arm system. The system is capable of creating smooth transients of MPs. Robustness to disturbances can be observed as local adaptations of individual low-level MPs, flexible sequencing of MPs, and global error recovery by changing the whole strategy of how to perform a movement skill.

I. INTRODUCTION

Several recent biological and neuroscientific studies suggest that animals as well as humans generate complex movement skills as combination 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.

Following the ideas of these findings, the present work suggests a similar control scheme for robot motion skills. Movement skills emerge by blending and sequencing of primitive motions, called movement primitives (MPs) in the remainder of this paper. It is the goal of the presented work to derive a control method that allows to consistently model the following properties: 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; easy integration of additional inputs like predicted future costs. Both low-level MPs and higher-level, coordinating control units responsible for sequencing and arbitration will be represented by dynamical systems.

The succeeding section will briefly introduce this work's notion of a movement primitive and how it is modeled as dynamical system. Section III focuses on the main topic of this paper, the coordinated activation of MPs to generate more complex 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 Section IV will illustrate the functional principles of the proposed methods. The paper closes with summary and outlook.

II. DEFINITION OF MOVEMENT PRIMITIVES

Movement primitives as understood in this paper are generating control commands in task or joint coordinates, and thus implement an elementary action, e. g. a reaching motion towards an object. Dynamical systems (DS) defined on the MP's state space are the method of choice to generate control commands in terms of changes of state:

$$\dot{\boldsymbol{y}} = f(\boldsymbol{y}, \boldsymbol{p}_{\mathrm{DS}}) \tag{1}$$

with $\boldsymbol{y} = s(\boldsymbol{x})$ selecting the subset of the full state $\boldsymbol{x}$ that is affecting the MP, and $\boldsymbol{p}_{\mathrm{DS}}$ being the parameter vector describing the behavior of the DS. This approach shows various benefits: due to the inherent state feedback, DSs are robust against disturbances, thus provide adaptation capabilities to the lowest level of the suggested control system; most common DS models are compact and have low computational cost for progression; DSs show good generalization capabilities; depending on the model used, DSs are time invariant.

Literature offers an increasing number of examples using DSs for motion representation. 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. Rather, several of the models mentioned above have been used. However, in addition to the DS generating control commands, 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 and to influence the activation of lower control levels (see next section). Furthermore, a prediction

[1]All authors are with Honda Research Institute Europe, Offenbach, Germany

[2]T. Luksch is with the Robotics Research Lab, University of Kaiserslautern, Germany t.luksch at cs.uni-kl.de

$\hat{\boldsymbol{y}}$ of the MP's state vector is generated based on the current state and control command. It is used to calculate a prediction error $p \in [0, 1]$ reflecting the difference of what is observed and what the MP is expecting to happen. Finally, each MP computes a goal distance $g \in [0, 1]$ and a responsibility value $r \in [0, 1]$ reflecting the current state. The 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 section.

The described MPs work independently of the underlying robot movement control. Corresponding to the robot used for the simulations in this work, the task-level control commands generated by the MPs are projected to joint torques $\boldsymbol{T}$ using a task-space inverse dynamics approach, see e. g. [12]:

$$\boldsymbol{T} = \boldsymbol{M}(\boldsymbol{q}) \left[\boldsymbol{J}^{\#}(\boldsymbol{a_y} - \dot{\boldsymbol{J}}\dot{\boldsymbol{q}}) - \boldsymbol{N}\boldsymbol{\xi} \right] + \boldsymbol{h}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{g}(\boldsymbol{q}) \quad (2)$$

The mass- and inertia matrix is denoted by $\boldsymbol{M}$. The activation values of the MPs enter the left Jacobian pseudo-inverse $\boldsymbol{J}^{\#} = (\boldsymbol{J}^T \boldsymbol{W_y} \boldsymbol{J} + \lambda \boldsymbol{I})^{-1} \boldsymbol{J}^T \boldsymbol{W_y}$ as diagonal elements of the weight matrix $\boldsymbol{W_y}$. Scalar λ is a small regularization value to increase numerical robustness. The task-level control law $\boldsymbol{a_y} = -k(\dot{\boldsymbol{y}}_{\text{curr}} - \dot{\boldsymbol{y}})$ is determined by the velocity error of the MP outputs of Eq. 1. Vector $\boldsymbol{\xi}$ is a gradient vector which accounts for joint speed damping and joint limit avoidance. It is projected into the null space of the movement through matrix $\boldsymbol{N}$. Vectors $\boldsymbol{h}$ and $\boldsymbol{g}$ comprise the gyroscopic and gravity terms.

Generalization capabilities and reusability of a MP is heavily influenced by the choice of task coordinates on which the MP operates. 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 DSs on such relative coordinates, no additional parameters have to be passed to MPs and the activation a can serve as the only input. Further details on choosing suitable frames of reference and their biological plausibility can be found in [13].

To clarify the MP interface, consider the example of 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 and clipped to $[0, 1]$. This way, g reflects the progression of the MP. 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 to be grasped 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. In using this prediction, major disturbances to the object or end effector pose would result in a large prediction error p.

In a similar way, the signals g, r, and p of all MPs are defined by simple equations to transform their state $\boldsymbol{y}$ to normalized values in $[0, 1]$ expressing the MP's interpretation of the current situation.

III. COMPOSING MOVEMENT SKILLS FROM MOVEMENT PRIMITIVES

In the following, movement skills shall be understood as a certain robot behavior emerging from a number of MPs coordinated by higher levels of the control hierarchy. In line with the findings in neuroscience mentioned above, such skills are 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 serial and concurrent activation of MPs. Additionally, it should be possible to adapt the activation stream, e. g. to recover from errors or to smoothly blend to another sequence of actions.

For this paper it is assumed that all MPs are already known either by designing them or by using imitation learning with state-of-the-art methods. Further, it is not the intention to create novel combinations of MPs by e. g. logical planning, instead skills are predefined. But the control system should be able to blend between already known MPs and skills based on sensory feedback, and the skill representation itself should allow adaptation to changes in the environment. The necessity of constantly having to replan should be avoided, as planning tends to be slow. State-of-the-art sampling-based planning algorithms still require a runtime of several seconds for complex robotic systems [14,15]. Also, planning often neglects experience from similar, previous situations, and is not biologically plausible on the "subconscious" level of behavior generation that is targeted in this paper.

The most common method that can be used to represent sequences 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 flows of actions.

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. The activation state will then continuously follow the new attractor behavior with a velocity depending on the time constant of the DS, thus generating smooth transients between MPs. This mechanism can uniformly create sequences as well as transitions of skill strategies, e. g. for error recovery. Furthermore, such an "activation dynamics" is exhibiting the same interface as the MPs described in Section II, i. e. it has an activation input and calculates signals p, g, and r. That way, high-level skills can recruit lower-level skills as well as MPs, creating a hierarchical, homogeneous control system.

Several approaches using a DS representation to activate control units can be found in literature. The first approaches emerged within the context of behavior-based robot control [16–19]. Specialized competitive dynamics are used to arbitrate behaviors. Parametrization of these dynamics allow to describe logical inter-dependencies between behavior resulting in the encoding of behavioral sequences.

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 remains unclear how well such monolithic systems will scale. Also, it is difficult to analyze the inner workings of the network as the encoding of the MPs and their interaction is distributed over all weights of the neural connections.

A few approaches use neural fields [20] as modeling tool for activations. While neural fields 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 [21,22]. Recently, there have been attempts to encode action sequences as DSs based on neural fields [23]. 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. concurrent MP activation or error recovery by skill transitions.

A. Activation Dynamics

The approach presented in the following is closely related to the work of Schöner and Steinhage cited above. While conceptually similar, it differs in the formulation of the dynamics. Instead of multiple coupled DSs with a tight integration of the actual sensor and control variables, a single DS is used that gains its power and flexibility by adding a number of external inputs as simple summation. Thus, it is less intermeshed with peculiarities of the robot and the scenario. Furthermore, the dynamics is easily extensible by adding additional inputs, e. g. predicted future costs as presented below.

The formulation of the dynamics used here to model activation dynamics (AD) is similar to the neural dynamics of continuous-time recurrent neural networks (RNN). While it resembles the neural field equations as used by Erlhagen, Schöner et al., it neglects the field characteristics as determining the activation of individual MPs does not require a continuous field with fixed lateral influence. Still, the RNN formulation features 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 [24]. In the following, the terminology of artificial neural networks will be applied to describe the characteristics of the DS.

Each MP that is controlled by the AD is represented by one neuron of the RNN as depicted by Figure 1. The dynamics of the potential u_i of neuron i follows the equation

$$\tau \dot{u_i} = -u_i - \kappa \cdot u_i^3 + h + w_0 \cdot f_\sigma(u_i) + s_i \qquad (3)$$

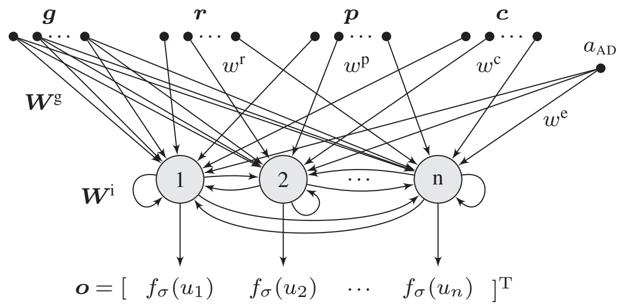

Fig. 1. Recurrent network used for the activation dynamics.

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:

$$f_\sigma(u) = \frac{1}{1 + e^{-\beta \cdot u}} \qquad (4)$$

The neural dynamics defined by Eq. 3 and 4 show a bistable behavior that is essential for its use as MP 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 s_i 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.

Without any external stimulus $s_i = 0$, the two stable fixed points occur at 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. One additional unstable fixed point exits between the two stable ones. External input s_i shifts the fixed points of the neural dynamics. When moved to very high or low values, e. g. due to very large inhibition, the transition time to a new fixed point caused by change of stimulus can be high. Depending on the dynamics time constant τ, this fact could slow down the reaction time to external disturbances. To counter this effect, Eq. 3 includes the cubic term $-\kappa \cdot u^3$ in addition to the standard formulation. A small value of κ will not drastically change the system's behavior, but will narrow the range of locations for the stable fixed points.

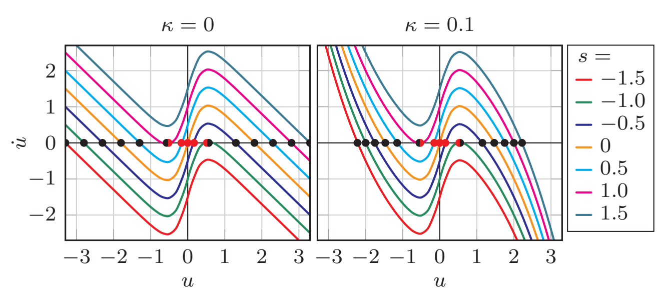

Fig. 2. Phase plot of neuron potential as in Eq. 3 for various stimuli s. Black dots mark stable fixed points, red dots mark unstable fixed points. ($h = -1.8$, $w_0 = 3.6$, $\beta = 5$. Left: $\kappa = 0$; Right: $\kappa = 0.1$)

Figure 2 illustrates the change of neuron potential given different external stimuli and the placement of the fixed points for $\kappa = 0$ (left plot) and $\kappa = 0.1$ (right plot). The zero-crossings with negative gradient of each line indicate stable fixed points and are marked by black dots. The unstable fixed point denoted by a red dot first converges and then vanishes with one of the stable fixed points as the stimulus increases or decreases, respectively, and the dynamics bifurcates.

The external stimulus s_i of Eq. 3 including the recurrent influence from other neurons is defined as

$$
\begin{aligned}
s_i = \quad & w^{\mathrm{e}} \cdot a_{\mathrm{AD}} && \text{(base excitation)} \quad (5)\\
& - \sum_{k,k\neq i} w^{\mathrm{i}}_{ik} \cdot f_\sigma(u_k) && \text{(recurrent inhibition)} \quad (6)\\
& - \sum_{j} w^{\mathrm{g}}_{ij} \cdot g_j && \text{(goal distance)} \quad (7)\\
& - w^{\mathrm{r}} \cdot (1 - r_i) && \text{(responsibility)} \quad (8)\\
& - w^{\mathrm{p}} \cdot p_i && \text{(prediction error)} \quad (9)\\
& - w^{\mathrm{c}} \cdot c_i && \text{(predicted costs)} \quad (10)
\end{aligned}
$$

Further terms introduce additional inhibition by considering the corresponding MP's responsibility r_i and its prediction error p_i, as well as requirements based on other MP's goal distances g_j. In addition, the activation $a_{\mathrm{AD}} \in [0, 1]$ of the controlling activation dynamics is added as base excitation.

The last part of the stimulus definition (Eq. 10) includes an exponentially decaying term c_i based on predicted future costs C:

$$
c_i = \begin{cases} \alpha \cdot c_i, & c_i > f_\sigma(u_i) \cdot C \\ f_\sigma(u_i) \cdot C, & \text{otherwise} \end{cases} \qquad (11)
$$

with $\alpha < 1$ and $C \in [0, 1]$. The global costs are calculated by internally simulating the robot's movement in the perceived environment within a receding future horizon of e. g. one second. The prediction starts from the current state of sensor feedback and activations, and progresses all dynamical systems as in regular control cycles. For each step of the prediction, joint limit and collision costs are accumulated as value C. In case of high resulting costs, for instance due to a predicted collision with an obstacle, the currently activated skill is inhibited by multiplying the costs with the neuron's output value $f_\sigma(u_i)$. The exponential decay sustains the inhibition to allow a competitive skill to be activated. Thus, an alternative strategy is chosen (for an example, see Section IV-B).

The last term (Eq. 10) of the stimulus definition has been added as an extension to an already working system. Previously set parameters did not have to be adapted. This illustrates the easy extensibility of the proposed formulation of the AD.

It is possible to express the dynamics of all neuron potentials of the RNN in a compact matrix notation which can be beneficial for implementation:

$$
\tau \dot{\boldsymbol{u}} = -\boldsymbol{u} - \kappa \boldsymbol{u}^3 + \boldsymbol{h} + w^{\mathrm{e}} \boldsymbol{a}_{\mathrm{AD}} + \boldsymbol{W}^{\mathrm{i}} \boldsymbol{o} + \boldsymbol{W}^{\mathrm{g}} \boldsymbol{g} + w^{\mathrm{r}} \boldsymbol{r} + w^{\mathrm{p}} \boldsymbol{p} + w^{\mathrm{c}} \boldsymbol{c} \qquad (12)
$$

with $\boldsymbol{u}^3 = (\dots, u_i^3, \dots)^{\mathrm{T}}$ denoting the cubic potential values, $\boldsymbol{h} = (\dots, h, \dots)^{\mathrm{T}}$ the resting levels, $\boldsymbol{a}_{\mathrm{AD}} = (\dots, a_{\mathrm{AD}}, \dots)^{\mathrm{T}}$ the activation of the AD, $\boldsymbol{o} = (\dots, f_\sigma(u_i), \dots)^{\mathrm{T}}$ the neuron's outputs, and $\boldsymbol{r} = (\dots, 1 - r_i, \dots)^{\mathrm{T}}$ the inverted responsibilities. Vectors $\boldsymbol{g}$, $\boldsymbol{p}$, and $\boldsymbol{c}$ hold the individual values g_i, p_i, and c_i, respectively. The diagonal of matrix $\boldsymbol{W}^{\mathrm{i}}$ is set to the self-excitation factor w_0. Figure 1 illustrates the resulting recurrent neural network using the weight notation of Eq. 12.

Considering the knowledge about the bifurcation behavior of Eq. 3, it is possible to find suitable parameters $w^{[\cdot]}$ to generate the desired behavior. For instance, given $h = -1.8$, $w_0 = 3.6$, and $\beta = 5$, the upper (lower) fixed point is destabilized for inputs $s_i < -1$ ($s_i > 1$). With a full base excitation of $a_{\mathrm{AD}} = 1$ and $w^{\mathrm{e}} = 2$, setting parameter $w^{\mathrm{r}} = 2$ would result in the neuron to become active for a responsibility of $r_i > 0.5$, and it would sustain that state even if the responsibility drops back to Zero. Similar considerations help to find the other parameters and design the desired behavior.

B. Sequences and Skill Transitions

Sequencing of MPs and transition between competing skills is achieved by choosing suitable weights of recurrent connections and of input signals. For sequences, the neuron representing MP$_i$ is inhibited by the output of neurons corresponding to those MPs succeeding MP$_i$ by setting the respective weights w^{i}_{ik} to negative values. Similarly, selecting among different skill options is accomplished by mutual inhibition of the neurons representing those skills. Thus, when the currently active skill is inhibited by e. g. a large prediction error (Eq. 9), 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 during the preceding time steps.

Beside the inhibition by recurrent connections, encoding of sequences makes use of goal distance values. A MP can be inhibited as long as preceding MPs are not close to their goal (Eq. 7). The goal distance is not necessary a spatial distance, but can be an arbitrary distance measure as the state of a MP could for instance also include force values.

As will be shown in the experiments below, a sequence can also be modeled to include concurrent activation of MPs by choosing appropriate values of $\boldsymbol{W}^{\mathrm{i}}$ and $\boldsymbol{W}^{\mathrm{g}}$.

IV. RESULTS

To illustrate the proposed movement generation method, this section describes several reach-and-grasp simulation experiments. The experiments make use of a simulated three-finger, 4 DoF Barrett Hand mounted on a 7 DoF Barrett WAM arm. The sophisticated physical simulation environment[1] allows to provide contact forces, joint torque feedback as well as measurements from the F/T-sensor installed at the wrist.

[1]The simulation environment is based on Vortex, a commercial physics engine by CMLabs (`www.vxsim.com`)

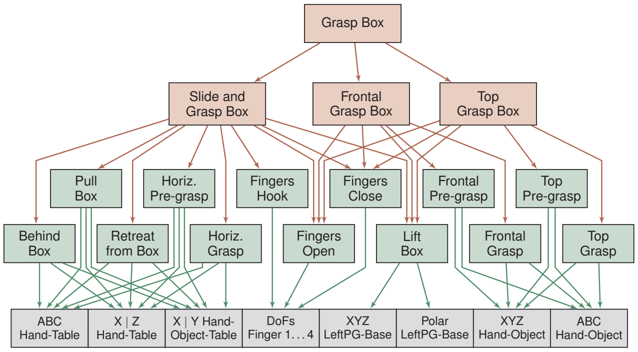

Fig. 3. Control hierarchy (green box: task coordinates, gray box: low-level MPs, red box: activation dynamics).

The control hierarchy used for the experiments is shown in Figure 3. For the sake of simplicity, all low-level MPs (green boxes) are modeled as linear attractor dynamics. They are defined on various task coordinates (gray boxes), e. g. relative hand-object transformation (XYZ Hand-Object). The high-level control units (red boxes) make use of the activation dynamics as described in the previous section. Red arrows indicate activation signals, green arrows depict task commands. Figure 4 depicts the simulation environment and image sequences of the three different grasping skills.

(a) Frontal Grasp Box

(b) Top Grasp Box

(c) Slide and Grasp Box

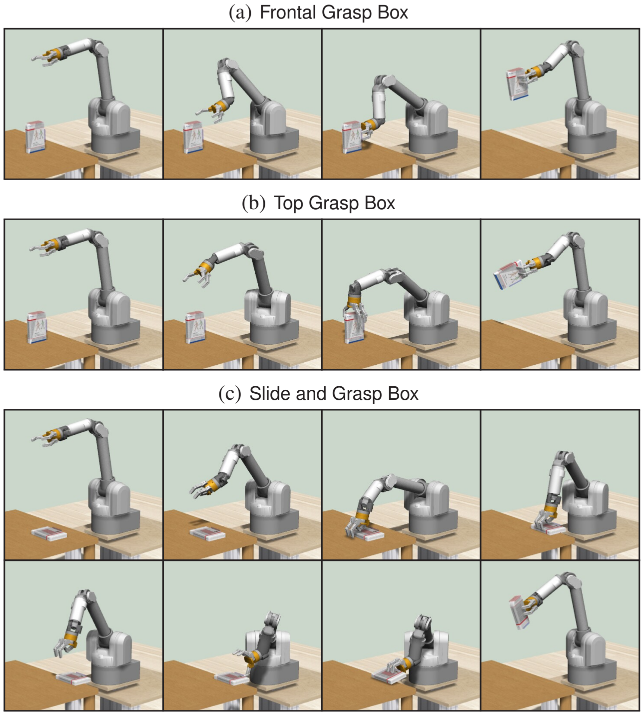

Fig. 4. Three different box grasping skills.

The following parameters of Eq. 3 were used: resting level is set to $h = -1.8$, self-excitation to $w_0 = -2 \cdot h = 3.6$, sigmoid function steepness to $\beta = 5$, and time constant to $\tau \in [0.1, 0.2]$. The factors governing the stimulus terms are set to $w^e = 2$, $w^r = 4$, $w^p = 4$, $w^c = 0$ for the Frontal, Top, and Slide-and-Grasp sequencing skills, and to $w^e = 2$, $w^r = 1.5$, $w^p = 4$, $w^c = 4$ for the competitive activation dynamics of Grasp Box. The latter features mutual inhibition by setting the non-diagonal elements of $\boldsymbol{W}^i$ to -4. Exemplary, Table I lists the MPs controlled by the Frontal Grasp Box AD and its weight matrices $\boldsymbol{W}^i$ and $\boldsymbol{W}^g$.

TABLE I

MPs CONTROLLED BY THE "FRONTAL GRASP BOX" SKILL AND ITS WEIGHT MATRICES FOR RECURRENT INHIBITION AND GOAL DISTANCES.

Name of MP	$\boldsymbol{W}^i$					$\boldsymbol{W}^g$				
Frontal Pre-Gr.	w_0	-4	0	-4	0	0	0	0	0	0
Frontal Grasp	0	w_0	0	0	-4	-0.8	0	-0.8	0	0
Fingers Open	0	0	w_0	-4	0	0	0	0	0	0
Fingers Close	-4	0	0	w_0	0	0	-1.5	0	0	0
Lift	-4	0	0	0	w_0	0	0	0	-1.5	0

A. Undisturbed Grasping Sequence

The progression of sequential and concurrent MPs shall be illustrated by a closer look at the internal and external signals of the grasping skill Frontal Grasp Box. Figure 5 visualizes the mechanisms of the activation dynamics during a grasping sequence without disturbances. The top-most plot depicts the vertical position of the book that is to be grasped. It is lifted from the table top starting at $t \approx 2.75$ 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.

The activation dynamics of the grasping skill is activated by higher levels of the control hierarchy at $t \approx 0.5$ sec, as can be seen by the increasing stimuli for all five neurons caused by the base excitation (Eq. 5). 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.

With the decline of the Pre-Grasp and Open MPs' goal distances the inhibition on the Grasp MP also decreases, allowing its potential to raise (Eq. 7). Its activation in turn causes an inhibition of the Pre-Grasp MP that is strong enough to terminate the activation of the neuron (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 uses the simulated measured grasping force as term in its calculation.

As demonstrated by this example, the activation dynamics is able to generate smooth transients between MPs during progressing a sequence. Additionally, concurrent activation of multiple MPs is supported. Local adaptation of disturbances is handled by the low-level MPs themselves, e. g. small to medium movements of the object during grasping would be compensated by the DSs of the Pre-Grasp and

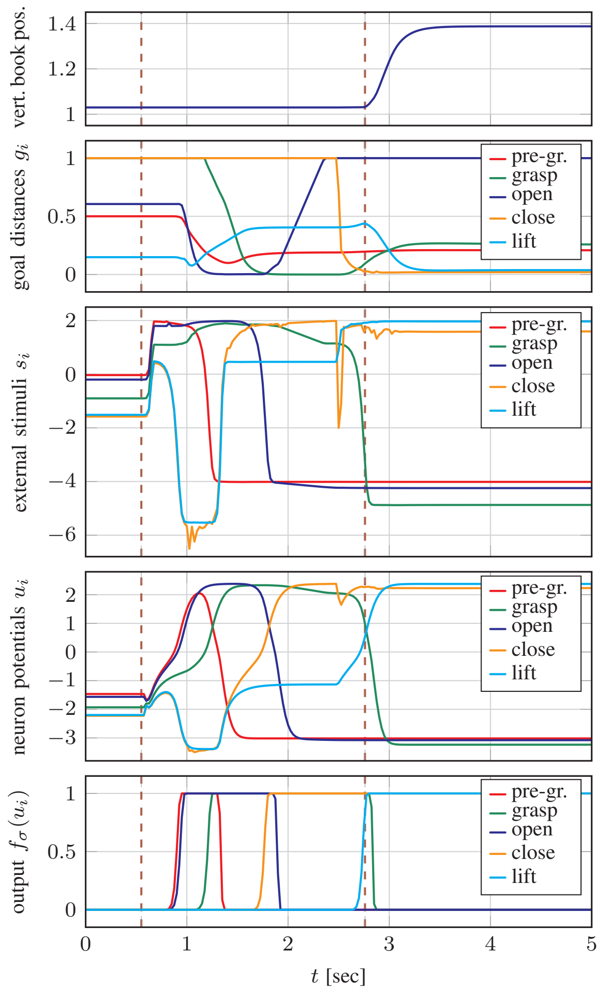

Fig. 5. Inputs, neuron potentials, and output values of the grasping skill's RNN during a run without external disturbances.

Grasp MPs[2]. This results from the continuous feedback of the DSs being defined on end effector coordinates relative to the object frame. 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. 9), causing the sequence to re-enter at an earlier point resulting in an emergent re-grasping behavior.

B. Transitions between Skills

Disturbances that cannot be handled by local adaptation or the sequence dynamics itself require higher-level adaptation. This can also be achieved by the proposed activation dynamics as implemented in the Grasp Box skill by using a competitive dynamics with mutual inhibition. When activated, this skill will choose that grasping variant among the Frontal, Top, and Slide-and-Grasp skills with the highest pre-activation. This preshaping is mainly determined by the corresponding responsibility and prediction error values.

[2]Low-level adaptation is also illustrated in the video supplement.

Figure 6 illustrates the case of the Frontal grasping skill reaching towards a book standing upright on the table (as in Fig. 4(a)). At $t \approx 1.2$ sec, a disturbance causes the book to fall over, creating a large prediction error of the grasping skill. Thus, the current skill is inhibited and the "next best" (see Section III-B) skill is selected, in this case the Slide-and-Grasp skill as the book is lying flat on the table. This skill then successfully finishes its sequence of MPs.

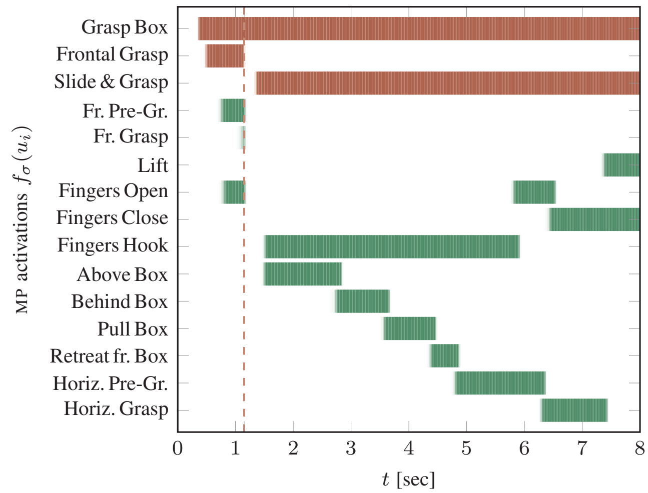

Fig. 6. Activations of various MPs involved in grasping (green) as well as of three higher-level ADs (red) during the falling book experiment.

A final experiment shows the effect of feeding the predicted collision and joint limit costs back to the activation dynamics. In this case, the robot is reaching towards a book standing on a table using the Frontal grasping skill as it has the highest responsibility. However, the space in front of the book is blocked by an obstacle. The top of Figure 7 illustrates how the predicted costs C increase as soon as the internally simulated robot collides with the obstacle within the future prediction horizon. The costs C cause an inhibition of the currently active Frontal grasping skill (decaying blue area in the stimulus plot). As soon as its activation decreases, the inhibition of the Top grasping skill is reduced and it is activated as alternative skill by the higher-level competitive dynamics of the Grasp Box skill.

This predictive mechanism allows the control system to recognize the infeasibility of the current movement strategy well before the actual occurrence of e. g. a collision event of the real robot. Further experiments have shown that the integration of joint limit considerations into the predicted costs calculation makes it possible to also detect future violations of the robot's working range when following the current movement dynamics.

V. SUMMARY AND OUTLOOK

This paper presents a movement generation architecture that is based on a hierarchical organization of dynamical systems. The novel aspects of this work lie in the neural dynamics-based organization of movement primitives. The proposed architecture is able to generate continuous sequences of possibly concurrently activated movement prim-

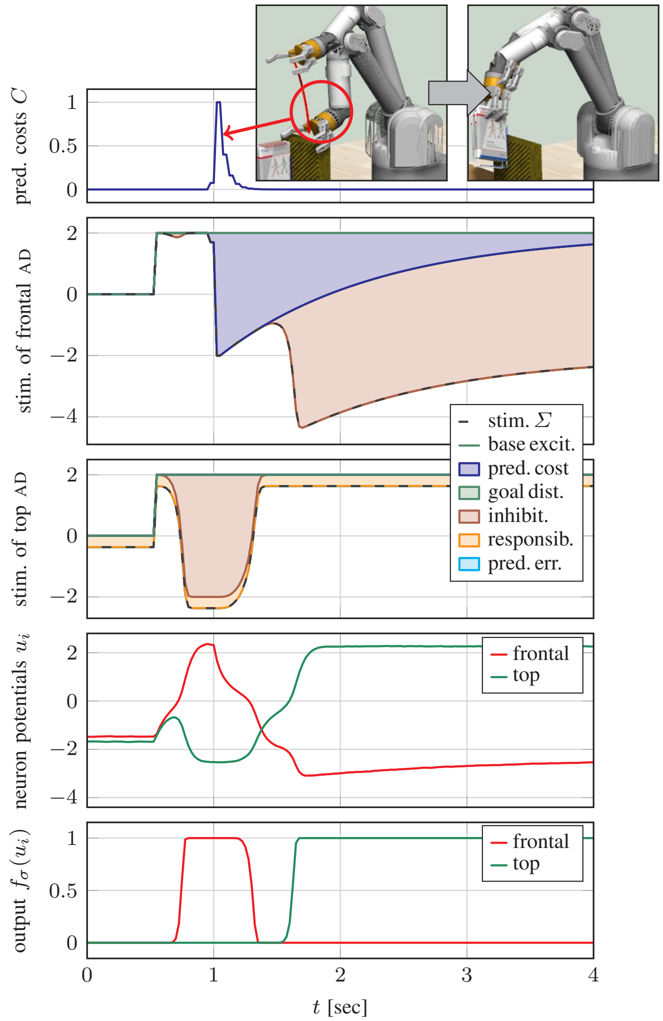

Fig. 7. A predicted future collision with an obstacle causes a transition between different grasping skills. The stimuli plots are broken down to the individual components as of Eq. 5–10.

itives with smooth transients. Its hierarchical organization makes it very scalable. Further, it features inherent failure recovery properties on several levels of execution: Local movement adaptation is realized by coupling the low-level dynamical systems output back to the movement control layer. Adaptation of the sequential behavior is achieved through competitive dynamics, allowing to change the sequential order of execution. And lastly, a predictive layer accounts for activating the most feasible skill in consideration of a receding horizon prediction.

A set of simulation results underlines the strength of the proposed concept. The parameterization of the system is intuitive and easy to do. Compared to discrete representations like FSMs, learning such systems is a parametric learning problem, which is less hard than structural learning. Future work will focus on this issue, in particular on the imitation learning of MPs and sequences as well as on the analysis of scenarios including force feedback with application on real 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. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot Modeling and Control*. John Wiley and Sons, 2006.

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

[14] D. Berenson, P. Abbeel, and K. Goldberg, "A robot path planning framework that learns from experience," in *Proc. of IEEE International Conference on Robotics and Automation (ICRA)*, 2012.

[15] A. S. K. Gochev and M. Likhachev, "Planning with adaptive dimensionality for mobile manipulation," in *Proc. of IEEE International Conference on Robotics and Automation (ICRA)*, 2012.

[16] G. Schöner, M. Dose, and C. Engels, "Dynamics of behavior: Theory and applications for autonomous robot architectures," *Robotics and Autonomous Systems*, vol. 16, no. 2, pp. 213–245, 1995.

[17] A. Steinhage and G. Schöner, "The dynamic approach to autonomous robot navigation," in *Proc. of IEEE International Symposium on Industrial Electronics (ISIE)*, 1997, pp. 7–12.

[18] 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.

[19] 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.

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

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

[22] 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.

[23] 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.

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