Elementary Dynamic Actions: key structures for contact-rich manipulation

Johannes Lachner[1], Moses C. Nah[1], Federico Tessari[1], Neville Hogan[1,2]

***Abstract*— Despite the slow neuromuscular system, humans achieve impressive dexterity in physical interactions. It is hypothesized that this is possible through "motor primitives," basic building blocks of motor control that enable dynamic behavior with minimal high-level control. In this paper, we review "Elementary Dynamic Actions" (EDA), which consist of three classes of motor primitives: submovements, oscillations, and impedance. We demonstrate how EDA can simplify robot control for contact-rich manipulation by implementing a peg-in-hole task on a real robot. Our assembly strategy remained unaffected by position errors of the insertion hole. Moreover, different workpiece shapes could be assembled for the same set of primitives. We briefly present our ongoing research to combine EDA with learning, reducing the numbers programmer-specified control parameters.**

I. INTRODUCTION

It appears to be a paradox that humans achieve such remarkable dexterity, despite their slow neuromuscular system. One hypothesis that can possibly resolve this paradox proposes that the human motor control system consists of basic building blocks, called “motor primitives” [1]. By composing control based on motor primitives, dynamic motor actions can be achieved without continuous intervention from higher levels of the Central Nervous System [2]. The concept of motor primitives dates back at least a century [3], with a number of subsequent experiments providing support for its existence in biological systems [4], [5], [6].

Motor primitives have been an inspiration for robot control. Two major control frameworks exist: Dynamic Movement Primitives (DMP) [7], [8], [9] and Elementary Dynamic Actions (EDA)[1] [1], [10], [11]. DMP uses kinematic motor primitives to generate the desired robot movement. It also provides a method called “Imitation Learning” to learn (or imitate) movements with arbitrary complexity [7]. EDA includes both kinematic and interactive primitives for robot control. The former parameterizes kinematic behavior and the latter manages interaction dynamics. A simple and heavily used implementation example of EDA is impedance control [12], [13], [14], which provides multiple advantages for tasks involving contact and physical interaction [15], [16].

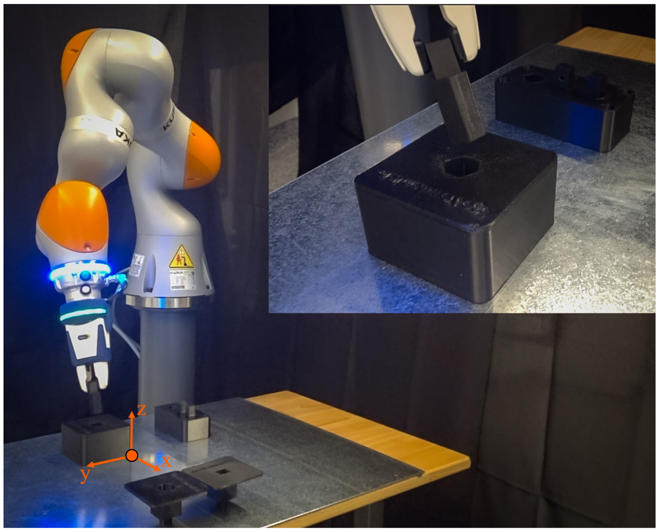

Fig. 1: Peg-in-hole assembly with the KUKA LBR iiwa.

Using a KUKA LBR iiwa 14 kg robot, we will show a peg-in-hole assembly task, involving three different workpiece shapes: a square, a hexagonal, and a circular peg (figure 1). A video of the assembly application can be found here: `https://youtu.be/T2pFu6IXk04`.

Our assembly strategy arose from industrial assembly applications with the KUKA LBR iiwa. The approach was robust against a displacement error of the insertion hole within ±3 mm along each planar direction (section III-A).

While using the same assembly strategy for all workpiece shapes, we discovered two notable findings that motivates our ongoing research (section III-B):

- The same set of impedance parameters worked for all three workpiece shapes. We therefore investigate the possibility of whether EDA enables a generalization of assembly strategies.
- Each task could be achieved within a range of impedance parameters. We currently investigate learning algorithms to find a range of feasible parameters, rather than one specific set of optimal impedance values.

The main contribution of this workshop paper is to present the theory of EDA and its applicability for contact-rich manipulation (section III-A). In section III-B, we will give an outlook of our ongoing research of impedance learning. With this approach, we aim to facilitate the programming of assembly tasks and reduce the number of parameters that have to be selected by the programmer.

II. ELEMENTARY DYNAMIC ACTIONS

EDA consists of three distinct classes of motor primitives: submovements and oscillations as kinematic primitives, and mechanical impedance as interaction primitive [10], [1], [11] (figure 2A). The three primitives can be combined by using the Norton equivalent network model (figure 2B) [17].

[1] Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, MA, USA.

[2] Department of Brain and Cognitive Sciences, Massachusetts Institute of Technology, Cambridge, MA, USA

[1] The original name suggested by Hogan and Sternad [1] was “Dynamic Motor Primitives.” Nevertheless, to avoid confusion due to the similarity to “Dynamic Movement Primitives,” here we use the term “Elementary Dynamic Actions” (EDA).

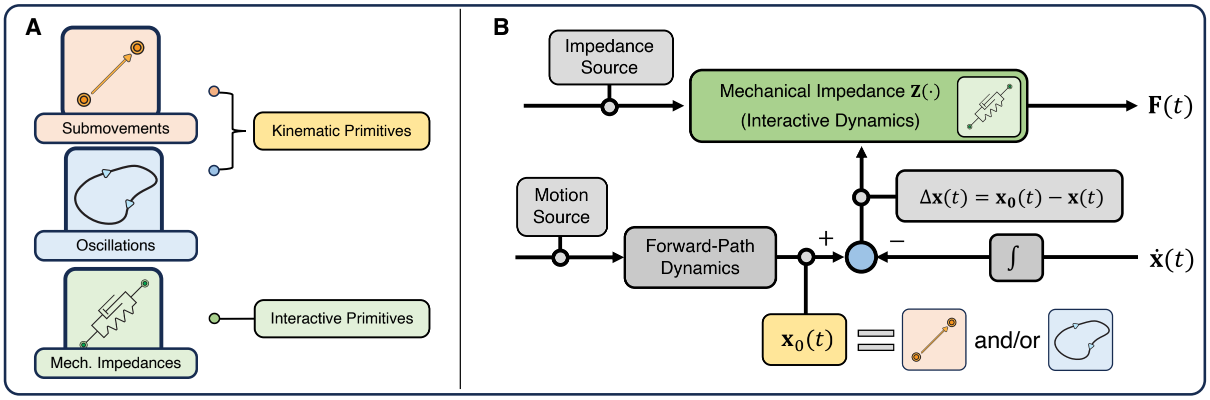

Fig. 2: (A) The three primitives of Elementary Dynamic Actions (EDA). Submovement and oscillation are kinematic primitives and mechanical impedance is an interactive primitive. (B) The three primitives are combined using a Norton equivalent network model.

For the remainder of the paper, we consider a system with n degrees of freedom (DOF).

A. Submovement

A submovement $\mathbf{x}_0 : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^n$ is a smooth trajectory in which its time derivative is a unimodal function, i.e., has a single peak value:

$$\dot{\mathbf{x}}_0(t) = \mathbf{v}\ \hat{\sigma}(t)$$

In this equation, $\hat{\sigma} : \mathbb{R}_{\geq 0} \rightarrow [0,1]$ denotes a smooth unimodal basis function with peak value 1; $\mathbf{v} \in \mathbb{R}^n$ is the velocity amplitude of the submovement. Submovements model discrete reaching motions, and therefore $\hat{\sigma}(t)$ has a finite support, i.e., there exists $T > 0$ such that $\hat{\sigma}(t) = 0$ for $t \geq T$. [18]

B. Oscillation

An oscillation $\mathbf{x}_0 : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^n$ is a smooth non-zero trajectory which is a periodic function:

$$\forall t > 0 : \quad \exists T > 0 : \quad \mathbf{x}_0(t) = \mathbf{x}_0(t+T)$$

Compared to submovements, oscillations model rhythmic and repetitive motions.

C. Mechanical Impedance

A mechanical impedance $\mathbf{Z} : \Delta\mathbf{x}(t) \longrightarrow \mathbf{F}(t)$ is an operator which maps (generalized) displacement $\Delta\mathbf{x}(t) \in \mathbb{R}^n$ to (generalized) force $\mathbf{F}(t) \in \mathbb{R}^n$ [10]. Here, $\Delta\mathbf{x}(t) \equiv \mathbf{x}_0(t) - \mathbf{x}(t)$ is the displacement between an actual trajectory of (generalized) position $\mathbf{x}(t)$ and a virtual trajectory $\mathbf{x}_0(t)$ to which the mechanical impedance is connected.

Along with the kinematic primitives (submovements and oscillations), EDA includes mechanical impedance as a distinct primitive to manage physical interaction [10], [19], [20]. The dynamics of physical interaction can be controlled by modulating mechanical impedance. For instance, tactile exploration and manipulation of fragile objects should evoke the use of low stiffness, while tasks such as drilling a hole requires high stiffness for object stabilization [19]. Moreover, impedance can be related to potential energy functions, which can be adapted to enable safe physical Human-Robot Interaciton [16].

Under the assumption that the environment is an admittance, mechanical impedance can be linearly superimposed even though each mechanical impedance is a nonlinear operator [14], [10]: $\mathbf{Z} = \sum \mathbf{Z}_i$.

D. Norton Equivalent Network Model

The three distinct motor primitives of EDA can be combined using a Norton equivalent network to model physical interaction [10] (figure 2B). The forward-path dynamics specify the virtual trajectory $\mathbf{x}_0(t)$, which consists of submovements and/or oscillations. The interactive dynamics, which consists of mechanical impedance $\mathbf{Z}$, determines the generalized force output $\mathbf{F}(t)$. The generalized displacement input is denoted $\Delta\mathbf{x}(t)$. Hence, a key objective of EDA is to find appropriate choices of $\mathbf{x}_0(t)$ and $\mathbf{Z}$ to produce the desired interactive behavior at the interaction port $(\mathbf{F}, \dot{\mathbf{x}})$.

As shown in figure 2B, EDA neither control $\mathbf{x}(t)$ (i.e., position) nor $\mathbf{F}(t)$ (i.e., force/torque) directly. This distinct feature of EDA has several benefits for robot control during physical interaction. Compared to $\mathbf{x}(t)$ and $\mathbf{F}(t)$ that depend on the environment or the object with which the robot interacts, $\mathbf{x}_0(t)$ and $\mathbf{Z}$ can be modulated independently, i.e., regardless of the environment or the manipulated object [19]. For instance, force control cannot be used for free-space motions and position control cannot be used in contact with the environment.

A common criticism is that EDA yields imperfections in terms of Cartesian accuracy. For most assembly tasks, however, we claim that Cartesian accuracy is not essential. In contrast, we show that EDA is robust against non-accurate workpiece locations.

E. Benefits of Elementary Dynamic Actions

1) Modular Framework for Robot Control: EDA provides modularity at the kinematic and dynamic level: submovements and/or oscillations can be overlayed at the level of virtual trajectory $\mathbf{x}_0(t)$ and multiple impedance can be superimposed. For instance, a submovement (i.e., discrete movement) can be directly overlayed with an oscillation (rhythmic movement). This modular property provides several benefits to simplify contact-rich manipulation, e.g., assembly tasks (section III-A).

The benefits of both motor-primitives approaches, DMP and EDA, can be combined. EDA provides a modular framework for robot control; DMP's Imitation Learning provides a rigorous mathematical framework to generate the virtual trajectory $\mathbf{x}_0(t)$. By combining these two approaches, each module can be learned separately and later combined, resulting in a "modular" Imitation Learning.

2) Passivity Property for Passive Environments: It is well known that robots capable of stable unconstrained motion may become unstable on contact with a physical constraint [21]. A well established approach to address this problem is ensuring the energetic passivity of the robot's interactive behavior [16], [22], [23], [24]. By superimposing mechanical impedance, passivity is preserved when interacting with passive environments [25]. The stability property of EDA provides several advantages for contact-rich manipulation task, e.g., peg-in-hole insertion tasks (section III-A).

3) Parallel Optimization of Impedance: As can be seen in the Norton Equivalent Network (figure 2B), the forward path dynamics and mechanical impedance are (separate) parallel network paths. Hence, $\mathbf{x}_0(t)$ and $\mathbf{Z}$ can be separately optimized, which has computational advantages for real-time control of robots with many DOF [26].

4) Simplified task-space control: The output force $\mathbf{F}$ of the impedance operational form [17] can simply be mapped to joint torques by using the transpose of the Jacobian matrix, which is beneficial for torque-controlled robots. Here, no inverse kinematics are needed [27], which enables seamless operation into and out of kinematic singularities [28]. For kinematically redundant robots, superposition of mechanical impedance can be used to manage kinematic redundancy without violating passivity [21], [24] and guaranteeing integrable motions without joint drift [29].

III. Application and Ongoing Research

A. Application: Peg-in-hole

For our assembly application, we used the KUKA Sunrise framework 1.17. Details about the algorithms behind the Cartesian Impedance controller can be found in [24]. Between the three workpieces and the respective insertion holes was a nominal clearance of 0.14 mm (square peg), 0.18 mm (hexagonal peg), and 0.20 mm (circular peg). All pieces were printed with a PRUSA i3 MK3 3D-printer using PLA filament. The components were printed with 20% infill with gyroid pattern.

Algorithm 1 shows the structure of the assembly application. For our application, we heavily used the torque sensors of the robot to predict external forces (Algorithm 1, line 3-5 and 14-20). All primitives of EDA were used and the controller specified the diagonal terms of the stiffness and damping matrices. A key for a successful assembly was to combine a rotational submovement with low rotational impedance (Algorithm 1, line 4-12). This strategy was motivated by observations of humans during assembly where motion and compliance is combined to assemble workpieces without even looking. With this approach, the peg automatically "slipped" into the hole, even though the workpiece was not perfectly aligned during the assembly. Due to the imperfections of the 3D-print, the riffles of the hole and the workpiece impeded the insertion of the peg. We therefore overlayed small oscillations to the discrete motion along $-z$ during the assembly (Algorithm 1, line 18-19). While certainly other assembly strategies would be successfull for this application (e.g., combining position control and perception), for us the key was the combination of tactile exploration with kinematic and dynamic primitives. This shows the benefits of the modularity of EDA.

Algorithm 1 Primitives for peg-in-hole assembly: submovements (orange), oscillations (purple), and mechanical Impedance (green). The coordinates can be seen in figure 1.

```
 1: Incline workpiece and align planarly at center of hole
 2: Create spatial Force Condition: $F_{\text{TCP}} < 15$ N
 3: while Force Condition do
 4:     Move linRel along $-x$ and $y$ coordinates
 5: end while
 6: Rotate from current TCP-pose to final TCP-pose, with
 7:     Siffness:
 8:     CartDOF.TRANSL $= 400$ [kg/s$^2$]
 9:     //Be compliant about A
10:     //for peg to "slip" into the hole
11:     CartDOF.A $= 5$ [kgm$^2$/s$^2$]
12:     CartDOF.B,C $= 100$ [kgm$^2$/s$^2$]
13:     Damping CartDOF.ALL $= 0.7$
14: while Force Condition do
15:     Move linRel along $-z$, with
16:     Siffness (line 8-12), Damping (line 13)
17:     //Overcome inaccuracy of 3D-print
18:     Overlay Oscillation about CartDOF.A, with
19:     Amplitude $A = 0.6$ [N], Frequency $f = 4.5$ [Hz]
20: end while
```

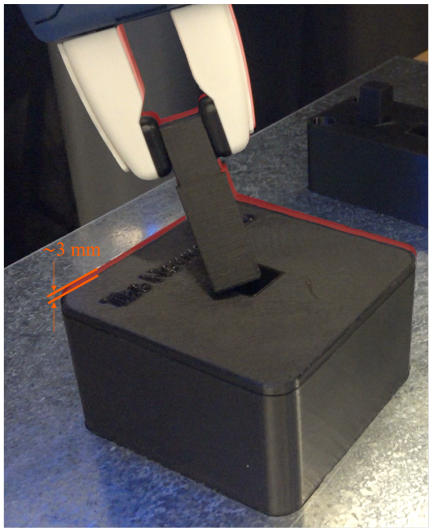

Fig. 3: Displacement tolerance during the peg-in-hole task.

In most manual production lines, there is a non-neglectable drift of the location of the conveyed workpiece. While control

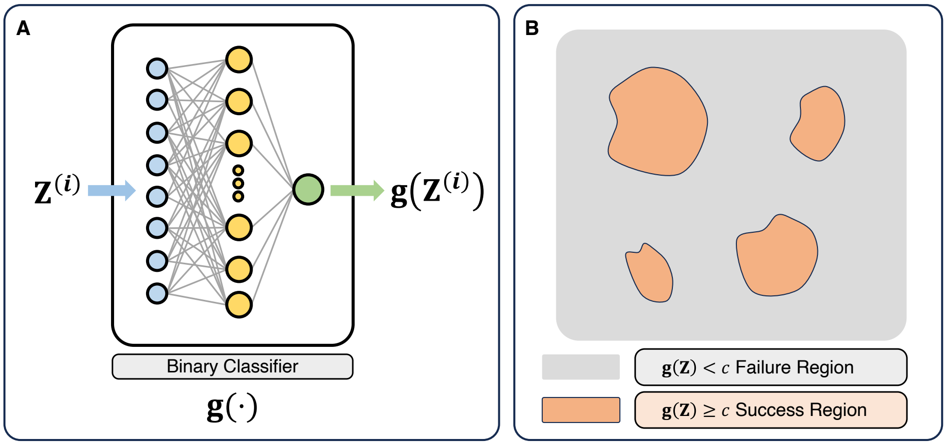
A
$\mathbf{Z}^{(i)}$
$\mathbf{g}(\mathbf{Z}^{(i)})$
Binary Classifier
$\mathbf{g}(\cdot)$
B
$\mathbf{g}(\mathbf{Z}) < c$ Failure Region
$\mathbf{g}(\mathbf{Z}) \geq c$ Success Region

Fig. 4: Binary classifier $\mathbf{g}_{\boldsymbol{\theta}}(\mathbf{Z})$, formulated as a neural network to predict whether a set of impedance parameters $\mathbf{Z}$ is success or failure.

algorithms like position control, force control, or hybrid force-position control struggle with this drift, our application was still successful for displacements errors of the insertion hole within ±3 mm (figure 3).

As can be seen in Algorithm 1, multiple parameters have to be chosen. The overall assembly strategy and the selection of impedance parameters was heuristically determined, based on previous experience in industrial applications. Less experienced programmers, however, might struggle to find a set of suitable impedance parameters.

The main goal of our current work is to minimize the parameters to be selected and therefore facilitate impedance programming. Our current approach tries to use learning methods to provide a range of feasible impedance values.

B. Learning feasible regions of impedance parameters

One common method of finding task-specific impedance parameters is to describe the task as an optimization problem [10], [30], [31], [32]. The impedance parameters are derived by optimizing a given cost function. While it is not trivial to find an appropriate cost function [33], a common choice includes the Cartesian error between the virtual and actual end-effector trajectory, i.e., the 2-norm of the displacement $\mathbf{x}_0(t)$ [32], [34]. The minimization of this cost functions results in one specific set of impedance parameters, which are often high impedance values on the matrix main diagonal [32].

For impedance-controlled robots, the authors challenge this common practice: 1) Using a cost function that involves the Cartesian error conflicts with the Norton equivalent network model (figure 2B) and does not take advantage of the parallel property between forward-path dynamics and mechanical impedance. 2) The output of most optimization and learning approaches is one single set of impedance, even though a range of parameters would be feasible. For example in our assembly strategy, changing the translational impedance parameters from 700 to 900 [kg/s^2] (Algorithm 1, line 8) would still be feasible, while changing the rotational impedance about the long side of the peg from 5 to 20 [kgm^2/s^2] (Algorithm 1, line 8) would not work.

In our current work, we aim to learn regions of impedance parameters that achieve the task. This formulation was motivated by Bernstein's philosophy *"repetition without repetition"* in human motor behavior [35].

We can identify regions of impedance parameters that result in success (1) or failure (0) by using binary classification methods [36], [37]. The benefit of binary classification methods is that it does not require a cost function design and can be readily combined with learning algorithms [38]. Using the same assembly strategy as for our peg-in-hole task (section III-A), we collected an initial data-set $\mathcal{D} := \{(\mathbf{Z}^{(i)}, y^{(i)})\}$ for $i \in \{1, 2, \cdots, N\}$, where $\mathbf{Z}^{(i)} = \{K_x^{(i)}, K_y^{(i)}, K_z^{(i)}, K_A^{(i)}, K_B^{(i)}, K_C^{(i)}, \xi_t, \xi_r\} \in \mathbb{R}^8_{>0}$ is an array of translational and rotaional stiffness parameters K_i and translational and rotational damping ratios ξ_i (algorithm 1); $y^{(i)} \in \{0, 1\}$ denotes success (1) or failure (0).

To collect the dataset $\mathcal{D}$, each element of the eight impedance parameters in $\mathbf{Z}^{(i)}$ is assigned three regions: low, medium, and high. Each value is then sampled in one of the regions, which results in 3^8 combinations. Considering that each trial requires $\simeq$ 20 seconds to determine failure ($y^{(i)} = 0$) or success ($y^{(i)} = 1$), all possible combinations require $20 \times 3^8 \approx 36$ hours of learning.

With this data-set, consisting of $N = 3^8$ sample points, we define a binary classifier function $\mathbf{g}_{\boldsymbol{\theta}}(\mathbf{Z})$, formulated as a neural network (figure 4A). Here, $\boldsymbol{\theta}$ denotes the parameters of the neural network. The weights of the neural network $\boldsymbol{\theta}$ are optimized with loss function $\mathcal{L}_{\boldsymbol{\theta}}(\mathcal{D}) = \sum_{i=1}^{N} \mathcal{L}(\mathbf{g}_{\boldsymbol{\theta}}(\mathbf{Z}^{(i)}), y^{(i)})$. For $\mathcal{L}$, we use binary cross-entropy loss function [39]:

$$\mathcal{L}(\mathbf{g}_{\boldsymbol{\theta}}(\mathbf{Z}^{(i)}), y^{(i)}) = \begin{cases} -\log(1 - \mathbf{g}_{\boldsymbol{\theta}}(\mathbf{Z}^{(i)})), & y^{(i)} = 0 \\ -\log(\mathbf{g}_{\boldsymbol{\theta}}(\mathbf{Z}^{(i)})), & y^{(i)} = 1 \end{cases}$$

Once the weights of $\mathbf{g}_{\boldsymbol{\theta}}$ are learned from the data-set $\mathcal{D}$, this binary classifier function is used to predict whether an arbitrary set of impedance parameters $\mathbf{Z}$ (not contained in $\mathcal{D}$) results in success or failure. Moreover, if one defines a threshold value $c \in \mathbb{R}$ for the function $\mathbf{g}$, regions of success (respectively failure) can be defined by finding the impedance values $\mathbf{Z}$ that results in $g(\mathbf{Z}) \geq 0$ (respectively $g(\mathbf{Z}) < 0$).

REFERENCES

[1] N. Hogan and D. Sternad, "Dynamic primitives of motor behavior," *Biological cybernetics*, vol. 106, no. 11, pp. 727–739, 2012.

[2] S.-W. Park, H. Marino, S. K. Charles, D. Sternad, and N. Hogan, "Moving slowly is hard for humans: limitations of dynamic primitives," *Journal of neurophysiology*, vol. 118, no. 1, pp. 69–83, 2017.

[3] C. S. Sherrington, *The integrative action of the nervous system*. A. Constable, 1906.

[4] N. A. Bernstein, "The problem of interrelation between coordination and localization." *Arch Biol Sci*, vol. 38, pp. 1–35, 1935.

[5] A. d'Avella, P. Saltiel, and E. Bizzi, "Combinations of muscle synergies in the construction of a natural motor behavior," *Nature neuroscience*, vol. 6, no. 3, pp. 300–308, 2003.

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

[7] S. Schaal, "Is imitation learning the route to humanoid robots?" *Trends in cognitive sciences*, vol. 3, no. 6, pp. 233–242, 1999.

[8] S. Schaal, J. Peters, J. Nakanishi, and A. Ijspeert, "Learning movement primitives," in *Robotics research. the eleventh international symposium*. Springer, 2005, pp. 561–572.

[9] A. J. Ijspeert, J. Nakanishi, H. Hoffmann, P. Pastor, and S. Schaal, "Dynamical movement primitives: learning attractor models for motor behaviors," *Neural computation*, vol. 25, no. 2, pp. 328–373, 2013.

[10] N. Hogan, "Physical interaction via dynamic primitives," in *Geometric and numerical foundations of movements*. Springer, 2017, pp. 269–299.

[11] N. Hogan and D. Sternad, "Dynamic primitives in the control of locomotion," *Frontiers in computational neuroscience*, vol. 7, p. 71, 2013.

[12] N. Hogan, "Mechanical impedance control in assistive devices and manipulators," in *Proc. of the 1980 Joint Automatic Control Conference*, 1980, pp. TA10–13.

[13] ——, "Impedance control: An approach to manipulation," in *1984 American control conference*. IEEE, 1984, pp. 304–313.

[14] ——, "Impedance control-an approach to manipulation. i-theory. ii-implementation. iii-applications," *ASME Journal of Dynamic Systems and Measurement Control B*, vol. 107, pp. 1–24, 1985.

[15] J. Colgate and N. Hogan, "Robust control of dynamically interacting systems," *International Journal of Control*, vol. 48, no. 1, pp. 65–88, 1988.

[16] J. Lachner, F. Allmendinger, E. Hobert, N. Hogan, and S. Stramigioli, "Energy budgets for coordinate invariant robot control in physical human–robot interaction," *The International Journal of Robotics Research*, vol. 40, no. 8-9, pp. 968–985, Aug. 2021.

[17] N. Hogan, "A general actuator model based on nonlinear equivalent networks," *IEEE/ASME Transactions on Mechatronics*, vol. 19, no. 6, pp. 1929–1939, 2013.

[18] N. Hogan and D. Sternad, "On rhythmic and discrete movements: reflections, definitions and implications for motor control," *Experimental brain research*, vol. 181, no. 1, pp. 13–30, 2007.

[19] N. Hogan and S. P. Buerger, "Impedance and interaction control," in *Robotics and automation handbook*. CRC press, 2018, pp. 375–398.

[20] N. Hogan, "Contact and physical interaction," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 5, pp. 179–203, 2022.

[21] J. Hermus, J. Lachner, D. Verdi, and N. Hogan, "Exploiting redundancy to facilitate physical interaction," *IEEE Transactions on Robotics*, vol. 38, no. 1, pp. 599–615, 2021.

[22] J. E. Colgate and N. Hogan, "Robust control of dynamically interacting systems," *International journal of Control*, vol. 48, no. 1, pp. 65–88, 1988.

[23] S. Stramigioli, "Energy-aware robotics," in *Mathematical control theory I: Nonlinear and hybrid control systems*. Springer, 2015, pp. 37–50.

[24] J. Lachner, "A geometric approach to robotic manipulation in physical human-robot interaction," Ph.D. dissertation, University of Twente, 2022.

[25] N. Hogan, "On the stability of manipulators performing contact tasks," *IEEE Journal on Robotics and Automation*, vol. 4, no. 6, pp. 677–686, Dec. 1988.

[26] J. Lachner, F. Allmendinger, S. Stramigioli, and N. Hogan, "Shaping impedances to comply with constrained task dynamics," *IEEE Transactions on Robotics*, vol. 38, no. 5, pp. 2750–2767, 2022.

[27] N. Hogan, "Stable execution of contact tasks using impedance control," in *Proceedings. 1987 IEEE International Conference on Robotics and Automation*, vol. 4. IEEE, 1987, pp. 1047–1054.

[28] D. Verdi *et al.*, "A compositional approach to robotic impedance control," Ph.D. dissertation, Massachusetts Institute of Technology, 2019.

[29] F. A. Mussa-Ivaldi and N. Hogan, "Integrable solutions of kinematic redundancy via impedance control," *The International Journal of Robotics Research*, vol. 10, no. 5, pp. 481–491, 1991.

[30] E. Theodorou, J. Buchli, and S. Schaal, "A generalized path integral control approach to reinforcement learning," *The Journal of Machine Learning Research*, vol. 11, pp. 3137–3181, 2010.

[31] J. Buchli, F. Stulp, E. Theodorou, and S. Schaal, "Learning variable impedance control," *The International Journal of Robotics Research*, vol. 30, no. 7, pp. 820–833, 2011.

[32] C. Yang, G. Ganesh, S. Haddadin, S. Parusel, A. Albu-Schaeffer, and E. Burdet, "Human-like adaptation of force and impedance in stable and unstable interactions," *IEEE transactions on robotics*, vol. 27, no. 5, pp. 918–930, 2011.

[33] A. Y. Ng, S. Russell *et al.*, "Algorithms for inverse reinforcement learning." in *Icml*, vol. 1, 2000, p. 2.

[34] X. Xiong, M. C. Nah, A. Krotov, and D. Sternad, "Online impedance adaptation facilitates manipulating a whip," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 9297–9302.

[35] N. Bernstein, "The coordination and regulation of movements," *(No Title)*, 1967.

[36] A. S. Therrien, D. M. Wolpert, and A. J. Bastian, "Effective reinforcement learning following cerebellar damage requires a balance between exploration and motor noise," *Brain*, vol. 139, no. 1, pp. 101–114, 2016.

[37] K. van der Kooij, N. M. van Mastrigt, E. M. Crowe, and J. B. Smeets, "Learning a reach trajectory based on binary reward feedback," *Scientific Reports*, vol. 11, no. 1, p. 2667, 2021.

[38] A. Singh, L. Yang, K. Hartikainen, C. Finn, and S. Levine, "End-to-end robotic reinforcement learning without reward engineering," *arXiv preprint arXiv:1904.07854*, 2019.

[39] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.