Synthesising a Motor-Primitive Inspired Control Architecture for Redundant Compliant Robots

Naveen Kuppuswamy[1] *, Hugo Gravato Marques[12], and Helmut Hauser[1]

[1] Institute for Informatics, A.I Lab,
University of Zürich, Switzerland
naveenoid@ifi.uzh.ch
[2] Department of Mechanical and Process Engineering, BIRL,
ETH Zürich,Switzerland

Abstract. This paper presents a control architecture for redundant and compliant robots inspired by the theory of biological motor primitives which are theorised to be the mechanism employed by the central nervous system in tackling the problem of redundancy in motor control. In our framework, inspired by self-organisational principles, the simulated robot is first perturbed by a form of spontaneous motor activity and the resulting state trajectory is utilised to reduce the control dimensionality using proper orthogonal decomposition. Motor primitives are then computed using a method based on singular value decomposition. Controllers for generating reduced dimensional commands to reach desired equilibrium positions in Cartesian space are then presented. The proposed architecture is successfully tested on a simulation of a compliant redundant robotic pendulum platform that uses antagonistically arranged series-elastic actuation.

1 Introduction

It has been argued that natural systems, in order to cope with uncertain, unstructured and dynamically changing environments evolve morphologies and material properties that are physically compliant (adaptable to external influences) and redundant (versatile in face of constraints), among other features [10]. The flip side of this argument is that the Central Nervous System (CNS) needs to cope with the large dimensionality thus induced. Even for simple end-point movements, a large number of muscles are recruited and, thus, have to be supplied with requisite input commands. Since the number of muscles is much higher than the number of variables in which the goal is defined, a single movement can be obtained by many different patterns of muscle activations; this is often referred to as Bernstein's *degrees of freedom problem* [3].

* This research has been supported by the European Community under the 7^{th} Framework Programme by the projects RobotDoc (Grant Agreement No. 235065), a Marie Curie Action ITN, and eSMCs (Grant Agreement No. 270212).

This problem is also important from the point of view of designing robotic systems for the real world. The problem of controlling dynamically complex systems is typically approached by explicitly using the (*inverse*) kinematic or dynamical models of the plant. However, the computational complexity drastically increases with the number of degrees of freedom [6]. Learning and optimisation theory offer an alternative to solving the redundancy problem. However, optimization algorithms suffer from an exponential increment of the computational complexity as a function of the dimensionality of the search space, the so called "curse of dimensionality" [12], rendering them intractable. Unsupervised learning methods [13] have also been proposed in this context, however their scalability to complex redundant and compliant robotic systems is unknown.

An alternative paradigm would be to look for a reduced dimensional specification of the system behaviour; a problem that has been studied in the domain of Model Order Reduction (MOR). MOR techniques aim at reducing the order of a dynamical system while preserving the input-output relationship to the extent possible [1]. The robot control applications of MOR techniques is largely underexplored, and we could take inspiration from nature in deriving techniques.

In this context, there is significant biological evidence suggesting that a process of dimensionality reduction may be occurring in neural control mechanisms [7]. The discovery of spinal Convergent Force Fields (CFFs) and their linear combinations in frogs [8] provided neurological justification for the presence of *motor primitives*, which have been described as fundamental units of the motor control system, suitable combinations of which enable complex movements to be carried out. Until now, most approaches have aimed to identify and model primitives from observations of natural movements.

In this paper, we propose a framework for synthesising a motor-primitive inspired control architecture for redundant and compliant robots. The architecture is inspired by recent work in biology [2], which proposed a novel model for the synthesis of motor primitives of a frog's leg using MOR and optimisation of a cost. The work we present adapts and expands their technique to artificial systems, and as a preliminary result we focus on linear dynamical systems. The results are demonstrated in a simulated tendon driven robotic pendulum which uses antagonistically arranged series-elastic actuation.

This paper is organised as follows. The considerations underlying the proposed architecture are presented in Section 2. In Section 3, the algorithm for extracting motor primitives is described. The experiments and simulated results are presented in Section 4, followed by the conclusions in Section 5.

2 Proposed Control Architecture - Considerations

Motor primitives have been characterised [2] as spinally stored constraints on the motor input commands of the form,

$$u = U^* C, \tag{1}$$

where, $U^* = u^*_{1\cdots k}$ is a set denoted as the motor primitives, comprising of k primitives, and C is the vector of reduced dimensional control inputs, $C =$

$[C_1, \dots, C_k]^T$. Each column of U^* can be thought of as a set of spinally stored muscle activations, similar to the activations produced by microstimulation of the frog's spine. In this formulation, each primitive u_i^*, is in the dimension of total number of muscles present [2], while only k primitives are needed and used to specify their motion. In this form, the primitives represent the basis vectors of the desired space of motor commands.

The approach of Berniker [2] to motor primitive synthesis used Balance Truncation [1], a control theoretic model reduction approach, to reduce the dimensionality of the mechanical system; the method relies on knowing the mechanical plant model. Furthermore, the approach assumes that motor primitives must be non-negative (real muscles cannot be negatively activated), orthogonal (act independently) and useful for generating commands (formalised based on mathematical properties of the equivalent reduced dimensional system); the primitive computation is based on optimising a corresponding cost function.

Ideally, in autonomous robots the architecture is synthesised in a self-organised manner, without knowledge of the full dimensional model in advance, which requires apriori system identification to be carried out. Also, some of the assumptions underlying the primitive synthesis approach [2] have to be adapted to comply with artificial actuation mechanisms. The technique proposed in this paper makes the following assumptions:

1. **Spontaneous motor activity is used to collect a dataset** : In order to self-organise a reduced dimensional model of the mechanical plant, spontaneous motor activity will be employed to perturb the system and collect a dataset characterising the behaviour, as described in Section 3.1.
2. **Statistical and data driven methods are used to reduce the dimensionality** : The Oja rule [9] demonstrated the ability of unsupervised learning in a network of neurons, to perform Principal Component Analysis (PCA). Hence for dimensionality reduction, a PCA based method, Proper Orthogonal Decomposition (POD), will be used as described in Section 3.2.
3. **Primitives can also be negative as motor commands can be negative for artificial systems** : For robotic systems, inputs may be negative as well, since typically most actuation mechanisms such as DC motors tend to exhibit bi-directionality at the output and bipolarity at the input. We address this consideration by proposing a technique for primitive synthesis using Singular Value Decomposition (SVD) described in Section 3.4.
4. **The reduced dimensional model is utilised to generate control inputs to reach equilibrium positions** : Motivated by the equilibrium point hypothesis [7] we propose a reduced dimensional controller that generates required motor commands to reach Cartesian space equilibrium positions as described in Section 3.5.

These new assumptions will be utilised in the synthesis of the control architecture as described subsequently. As a preliminary exploration we shall constraint our proposal to linear dynamical systems, although it can potentially be adapted to nonlinear systems as well.

3 Synthesis Methodology

The proposed reduced dimensional architecture is synthesised using the methodology presented in Fig. 1. The various constituent processes are described in this section.

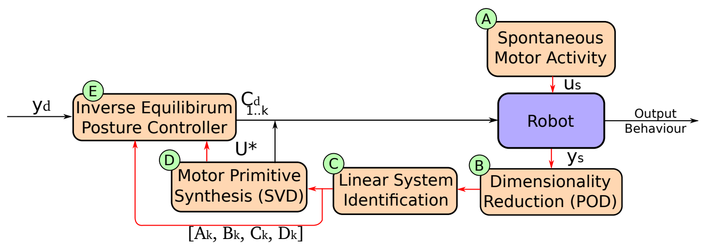


Fig. 1. Motor primitive-inspired Control Architecture - Synthesis Methodology. First the Robot is perturbed by Spontaneous Motor Activity $u(t)$ *(A)* to generate a dataset y_s and subsequently MOR (POD) *(B)* and Linear System Identification *(C)* are applied to yield a reduced order model. Primitives are then synthesised using SVD *(D)* and are combined with the reduced model in the equilibrium posture controller *(E)* to generate motor commands corresponding to desired behaviour goals in end-effector space.

3.1 Dataset generation through spontaneous motor activity

In mammals, the process of spontaneous motor activity (SMA) carries out muscle contractions in the absence of sensory stimulation. This type of motor activity has been observed during sleep throughout all developmental stages (including the foetal stage) [5]. One particular type of SMA observed is the Myoclonic twitch which spontaneously triggers independent contractions of individual muscles. Inspired by this process, we utilise independent and individual pulse inputs $u_s(t)$ (square signals of amplitude 0.01 and duration $2.8s$) to perturb the mechanical system, resulting in a motion output that can be recorded in the form of a dataset (see block *(A)* in Fig.1)of *snapshots* of the dynamical system as $\chi = [x(t_0), \ldots, x(t_i)]$, where $\chi \in \mathbb{R}^{N \times n_t}$ and $x(t_i)$ is the ${n_t}^{th}$ snapshot of the system, where n_t is the total number of snapshots in the dataset (or datapoints) and N is the state dimensionality.

3.2 Reduction using Proper Orthogonal Decomposition (POD)

The next step is to reduce the dimensionality of the dataset using POD[3] as depicted in block *(B)* in Fig.1. Consider a linear dynamical system of the form below,

$$\dot{x} = Ax + Bu, \quad y = Cx, \tag{2}$$

[3] also called PCA, Karhunen-Loeve decomposition or factor analysis

where, $u \in \mathbb{R}^I$ is the input, $x \in \mathbb{R}^N$ is the state, $y \in \mathbb{R}^O$ is the output. The matrices A, B, and C are commonly called as the state, input, and output matrices, respectively. In this case, the reduction aims to find a lower dimensional representation z such that,

$$\dot{z} = A_k z + B_k u, \quad y = C_k z + D_k u, \tag{3}$$

where, $z \in \mathbb{R}^k$ is the state, and D is a feedthrough matrix compensating for steady state differences. We thus look to replace the N dimensional system by a nearly-equivalent (similar in behaviour) k dimensional system, where $k \ll N$. From the dataset χ collected in the previous stage, the Singular Value Decomposition (SVD) then can be used to obtain the *best* [11] reduced dimensional approximation $\hat{\chi}_k$ which minimises the norm $\| \chi - \hat{\chi}_k \|_2$. The SVD renders $\chi = U\Sigma V^T$, where $UU^T = I, VV^T = I$, and the singular values are ordered as $\sigma_1 \geq \ldots \sigma_k \geq \ldots \sigma_n$. We can then truncate χ to the first k singular values, by using the corresponding first U_k singular vectors as a basis of the k-dimensional subspace we are projecting the dataset to, as $z(t) = U_k x(t)$. The next step is to obtain the model parameters.

3.3 Identification on the reduced dimensional dataset

To identify the reduced dimensional model, we employ system identification on the dataset $z(t)$ as depicted in block *(C)* in Fig.1 . Due to the assumption of linear dynamics, the dataset $z(t)$ obtained from POD will also be guaranteed to be linear [1] and linear least squares identification can be employed as,

$$[\dot{z}(t)] = [A_k, B_k][z^T(t), u^T(t)]^T, \quad [y(t)] = [C_k, D_k][x^T(t), u^T(t)]^T, \tag{4}$$

where, z is the new state variable of the dynamical system, A_k, B_k, C_k, and D_k are the reduced dimensional state, input, output and feedthrough matrices respectively. Note that $u(t)$ and $y(t)$ have not changed from the original system in Eq. 2.

3.4 Primitive Synthesis using SVD

Once the reduced order model is obtained, primitives in the form of Eq. 1 are computed. An important criterion for the primitives is that ideally the commands generated in the reduced dimensional space are "useful" in the sense of their effect on the state [2]. This is ensured by allowing the primitives U^* to be orthogonal to the nullspace of the reduced dimensional input matrix B_k. Moreover, the primitives are orthogonal to each other (to allow spanning the control input space). Both these goals are accomplished by finding the singular vectors of B_k and choosing the last k of these vectors.

Consider the singular value decomposition of a matrix B_k, $B_k = U\Sigma V^*$. The null space of the input matrix B_k has as its basis, the last $n - k$ columns of the right singular vectors V^* of the decomposition [11]. Since the columns of the V^*

matrix are orthogonal to each other, the first k columns thus can be chosen as primitives, since they are both useful and orthogonal.

$$u^* = \mathcal{N}(B_k)^{\perp}, \quad u^* = V^*_{1\ldots k}, \tag{5}$$

where the operator $\mathcal{N}()^{\perp}$ computes the nullspace complement of a matrix. Due to the availability of multiple high speed numerical SVD computing algorithms, primitives computation is faster than methods based numerical optimisation [2].

3.5 Feedforward Equilibrium Posture Controller Design

Once the primitives are computed, a controller can be designed as required. In [4] and [8], it is suggested that the controller is feedforward in structure and it generates the necessary commands to affect the equilibrium posture of the limb. For the obtained system in Eq. 3, the equilibrium state for a given input corresponds to $\dot{z} = 0$, which is therefore,

$$z = -A_k^{-1} B_k u, \quad y = \left[-A_k^{-1} B_k + D_k\right] u. \tag{6}$$

Since the input u is constrained according the motor primitive as in Eq.1, it is sufficient to compute the required reduced dimensional control inputs C_d for a desired output y_d where $C_i \in \mathbb{R}^k$. For this, the pseudo-inverse or the Moore-Penrose inverse (†) can be used to obtain,

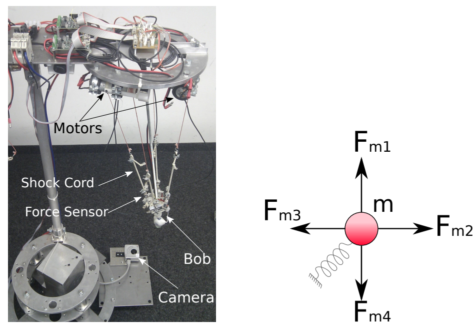
$$C_d = \left[\left(-C_k A_k^{-1} B_k + D_k\right) u^*\right]^{\dagger} y_d. \tag{7}$$

Note that if k is chosen to be of same dimensionality of the output y, Eq.7 is computed using a regular inverse instead of the pseudo-inverse and thus the redundancy problem is directly resolved.

4 Experiments and Results

4.1 Methods : Pendulum Robot and Simulation

The pendulum robot platform is a test setup built to investigate methods and techniques for developmental robotics. Loosely inspired by the human shoulder system, it consists of two mechanically independent pendula, each driven by 4 series elastic actuators coupled in an agonist-antagonist configuration, as shown in Fig.2a. Each muscle system can be actuated independently and includes force and elongation sensors. A camera is mounted on the base of the pendulum looking upwards to extract end-point position as a 2D position measured in the camera frame of reference. The dynamics of this robot is be assumed to be linear under the conditions of bounded amplitude motion due to the relatively long length of the muscles. Since the platform is driven by 4 motors, the input dimensionality is 4. Thus in order to be interesting, we must synthesise a controller with k primitives where $1 < k \leq 4$ to perform meaningful tasks in the 2D task space.



(a) (b)

Fig. 2. a) Pendulum robot platform and b) Linear System simulation of the Pendulum robot platform (the mass is that of the end-point bob)

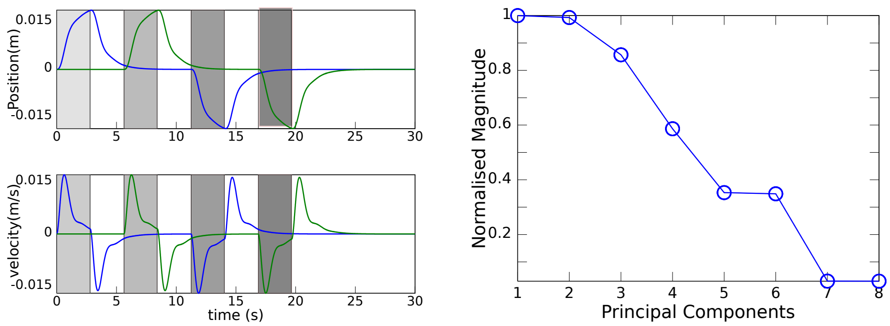
The simulator depicted in Fig.2b uses a linear approximation of the plant. The nonlinearities due to angles of force application are neglected as a simplification. The simulation implements the following model,

$$
\begin{aligned}
\ddot{x}_c &= -k_x x_c - b_x \dot{x}_c + \sum_{i=1}^{4} F_{m_i} cos(\theta_i), \\
\ddot{y}_c &= -k_y y_c - b_y \dot{y}_c + \sum_{i=1}^{4} F_{m_i} sin(\theta_i), \\
\dot{\alpha}_i &= \tau(u_i g - \alpha_i), \quad F_{m_i} = k_m \alpha_i,
\end{aligned}
\tag{8}
$$

where, $k_{x,y}$ is the stiffness of restoring force to mean position, $b_{x,y}$, is the damping of the pendulum bob, $i \in [1,4]$ $\tau_{1 \cdots 4}$ are the time constants of the muscle (critically damped), $g_{1 \cdots 4}$ are the gains on the input signal, k_m is the muscle stiffness proportionality to its activation, and θ_i is the mounting angle of each of the spindle motors. Note that, x_c and y_c in this model are both state variables and should not be confused with the state x and output y of Eq.2. For this paper, the simulation constants were fixed as $k_{x,y} = 10$, $b_{x,y} = 5$, $\tau = 1$, $g = 5$, and $k_m = 5$ (currently being validated on the real robot platform). The model has dimensions 8 on state and 4 on input, corresponding to desired angular positions on DC motors with controllers of time constants τ and dc gain $g_{1 \cdots 4}$. The model was implemented in GNU Octave and integrated using the *ODE45* routine

4.2 Results - Spontaneous Motor Activity and Dimensionality Reduction

A unit pulse input applied to each muscle sequentially to replicate the spontaneous motor activity in the form of single muscle twitches as shown in Fig.3a. The various state and output trajectories $y_s(t)$ and $x_s(t)$ respectively, were recorded



(a) (b)

Fig. 3. a) Single Muscle Twitching and output of the simulated robot and b) Principal components of the dataset $(y_s(t))$ -Scree Plot

and stored as a dataset and POD was used to reduce the dimensionality. The principal components are depicted in the scree plot in Fig.3b. The first k largest components were chosen to compute controllers with k motor primitives, where $1 < k \leq 4$, thus giving 3 types of controllers. Linear dynamical models of dimension k were then obtained by fitting in each case.

4.3 Results - Synthesised Motor Primitives and task performance

From the identified reduced dimensional linear state space, the primitives were synthesised using SVD for the cases of $k = 2, 3$, and 4 primitives. The computed primitives in each case are depicted in Table 1. The synthesised primitives are visualised by locating the resulting equilibrium points (at $\dot{x} = 0$) for a unit inputs applied to each of the reduced dimensional input C individually. Since the source system is linear, the equilibrium points obtained are unique and independent of the initial conditions as depicted in Fig. 4. Since any position in the Cartesian space can be obtained by using the right input C, the knowledge of these equilibrium points can be used to generalise to new points in the task space by using linear combinations similar to the biological case [8].

A reaching task was then computed using the controller form of Eq. 7 for a set of 3 controllers ($k = 2, 3, 4$), as shown in Fig. 5. In each case small offset errors result in steady state due to the quality of the obtained reduced dimensional model. This offset error could potentially be minimised if the model can be

$k=2$		$k=3$			$k=4$			
-0.53210	-0.49997	-0.56734	-0.41926	0.49501	-0.55065	-0.44407	-0.50324	0.49632
0.46796	-0.49313	0.56418	-0.56742	-0.34017	0.57413	-0.53848	0.34869	0.50874
0.54176	0.41663	0.43817	0.57504	0.48379	0.40476	0.59823	-0.47541	0.50227
-0.45208	0.57730	-0.40967	0.41423	-0.63655	-0.45091	0.39365	0.63178	0.49252

Table 1. Computed U^* for the cases of 2, 3, and 4 primitives

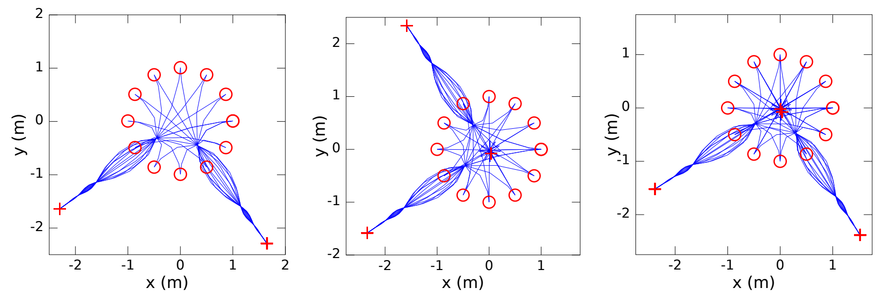


Fig. 4. Equilibrium positions of endpoint (*red* stars) while using (a) $k = 2$, and individual unit inputs in C i.e. $C_1 = 1, C_2 = 0$, and vice versa (b) $k = 3$, and individual unit inputs in C as before, and (c) $k = 4$, and individual unit inputs in C as before, the initial conditions (*red* circles) are chosen to lie in a circle about the center. The trajectories of the endpoints are in *blue* lines. In cases (b) and (c) the latter equilibrium points are are found to lie nearly at the origin of the workspace.

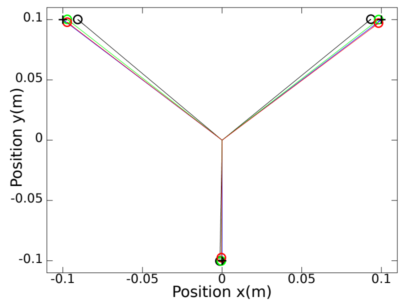
improved through subsequent stages of learning and adaptation. More complex desired trajectories using multiple waypoints can also be obtained using the controller as shown in Fig. 5.

5 Conclusions

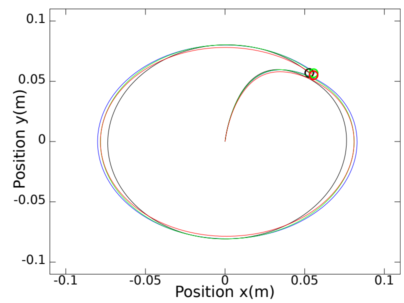
This paper presented a motor primitive inspired architecture for reduced dimensional control of redundant compliant robots. Based on a biological model of motor primitives using model order reduction, considerations relevant to artificial system control were presented. A technique for self-organising a controller was presented, inspired by the concept of spontaneous motor activity. A reduced dimensional representation of the ensuing dataset was then used to synthesise motor primitives using SVD. The computed primitives were then utilised to compute the necessary control, across all of the inputs, for reaching fixed points in space. The proposed framework was tested on simulated version of a compliant redundant tendon driven robot platform. The preliminary simulation based results are promising and demonstrate the utility of the proposed technique for application to artificial systems. From an engineering viewpoint an extension of the work to the nonlinear systems such as kinematic chains is currently being carried out. An important consequence for biological systems arising as an extension of this work is an investigation on the relationship between dimensionality reduction and mechanical properties of biological systems.

References

1. A. Antoulas, D. Sorensen, and S. Gugercin. A survey of model reduction methods for large-scale systems. *Contemporary Mathematics*, 280:193–219, 2001.



(a)



(b)

Fig. 5. Performance of the 3 controllers, $k = 2$ (*red*), $k = 3$ (*green*) and, $k = 4$(*black*) relative to an ideal controller (*blue*), in performing (a) reaching task in Cartesian Space to the positions $(0.1, 0.1)$, $(0, -0.1)$, and $(-0.1, -0.1)$, (b) continuous tracking task in Cartesian Space to a circle centered at $(0, 0)$ and diameter 0.15m. The trajectories obtained in each case are nearly identical.

2. M. Berniker, A. Jarc, E. Bizzi, and M. C. Tresch. Simplified and effective motor control based on muscle synergies to exploit musculoskeletal dynamics. 106:7601–6+, 2009.
3. N. Bernstein. *The Co-ordination and Regulation of Movements*. Oxford, UK: Pergamo, 1967.
4. E. Bizzi, F. A. Mussa-Ivaldi, and S. Giszter. Computations underlying the execution of movement: a biological perspective. *Science*, 253(5017):287–291, July 1991.
5. M. S. Blumberg and D. E. Lucas. Dual mechanisms of twitching during sleep in neonatal rats. *Behav Neurosci*, 108:1196–1202, 1994.
6. R. Featherstone. *Rigid Body Dynamics Algorithms*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2007.
7. M. L. Latash. Evolution of motor control: From reflexes and motor programs to the equilibrium-point hypothesis. *Journal of human kinetics*, 19(19):3–24, 2008.
8. F. Mussa-Ivaldi, A. Giszter, and E. Bizzi. Linear combination of primitives in vertebrate motor control. *PNAS USA*, 91:7534–7538, 1994.
9. E. Oja. Simplified neuron model as a principal component analyzer. *Journal of Mathematical Biology*, 15:267–273, 1982. 10.1007/BF00275687.
10. R. Pfeifer, M. Lungarella, and F. Iida. Self-organization, embodiment, and biologically inspired robotics. *Science*, 318(5853):1088–1093, 2007.
11. G. Strang. *Linear Algebra and Its Applications*. Brooks Cole, February 1988.
12. R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, 1998.
13. E. Todorov, W. Li, and X. Pan. From task parameters to motor synergies: A hierarchical framework for approximately-optimal control of redundant manipulators. *Journal of Robotic Systems*, 22(11):691–710, 2005.