INVITED PAPER

A Comprehensive Realization of Robot Skin: Sensors, Sensing, Control, and Applications

This article presents a holistic approach to engineer the artificial skin for robots with an example of a multimodal skin cell showing multiple humanlike sensing modalities.

By Gordon Cheng, *Fellow IEEE*, Emmanuel Dean-Leon, *Member IEEE*, Florian Bergner, *Student Member IEEE*, Julio Rogelio Guadarrama Olvera, *Student Member IEEE*, Quentin Leboutet, *Student Member IEEE*, and Philipp Mittendorfer, *Member IEEE*

ABSTRACT | This article presents a holistic approach to the engineering of an artificial robot skin for robots. An example of a multimodal skin cell is given, one that supports multiple human-like sensing modalities, and support for skin cell network is also provided; this is essential to form large-area skin patches in order to cover the surfaces of robots. The essential elements of efficiently handling a large amount of tactile data are explained. A general control framework, which supports robots commanded in position, velocity, and torque, is provided and validated. Several applications of this robot skin will be presented, demonstrating the effectiveness and efficiency of our artificial robot skin to support a wide number of robotic platforms as well as its ease of use across different domains.

KEYWORDS | Artificial robot skin; robot skin interactions; skin-based robot control

I. INTRODUCTION

This article presents the engineering of an artificial robot skin for robots, and it will provide a step-by-step guide to the development of an artificial robot skin system that targets a complete solution that is efficient, flexible, scalable, and robust. Achieving these attributes will surely enable the deployment of an artificial robot skin on any autonomous robot (see Fig. 1).

An artificial robot skin on robots has been of interest since the early days of robotics [1], [2]. Many of the considerations at that time, and still now, were to enable robots with a sense of touch similar to that of humans. This allows robots to be fully physically interactive and effectively safe to be near humans.

We set out to minimize possible errors in several aspects and advocate an automated approach for the topological and spatial self-configuration of each skin cell and skin patch, and the automatic extraction of robot parameters. Moreover, we champion the importance of the organizational aspects of tactile information, in a way that effectively reduces overheads in communication and in the processing of sensory information on a central computer.

Furthermore, we strongly believe that a solution needs to support all standard robot control interfaces (i.e., position, velocity, and torque), thus providing a fully adaptable solution for almost any robot.

A. Related Work

1) Robot Skin: The development of the robot skin that provides adequate tactile sensing capabilities for

Manuscript received February 27, 2019; revised July 3, 2019; accepted July 27, 2019. Date of publication August 26, 2019; date of current version October 3, 2019. This work was supported by the Deutsche Forschungsgemeinschaft (German Research Foundation) (DFG) through the Major Research Instrumentation Program in 2015. *(Corresponding author: Gordon Cheng.)*

The authors are with the Institute for Cognitive Systems, Technical University of Munich, 80333 Munich, Germany (e-mail: gordon@tum.de).

Digital Object Identifier 10.1109/JPROC.2019.2933348

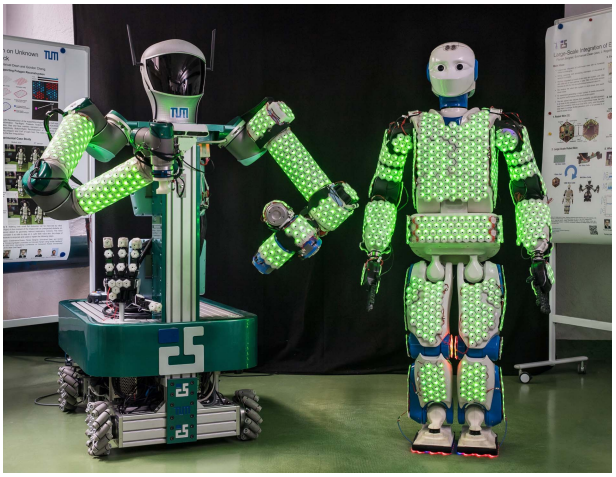


Fig. 1. *Different robots using our robot skin: The Allegro Hand covered with 62 skin cells; the TOMM robot with dual arms covered with 700 skin cells; and the H1 robot whose whole body is covered with 1260 skin cells. All these robots use different command interfaces: torque, velocity, and position commands, respectively. (Photo credit: A.Eckert/TUM.)*

different robot applications is still an ongoing process, and a wide variety of potential solutions have emerged. The development of such robot skin systems is challenging since it involves the organization and calibration of a large number of spatially distributed discrete sensors and the handling of a large amount of information. A review and comparison of tactile sensing in robotics has been conducted in [3]. This article summarizes the requirements and design principles for evolving robot skin systems toward the capabilities of the human skin. The key challenges, such as reliability, robustness, simple deployment, wiring, feasible acquisition of sensor poses, and efficient and low-latency transmission and processing of a large amount of tactile information, are discussed in [4]. An earlier development toward a robot skin system has been shown in [5] and [6]. The proposed system utilizes the arrays of IR-based proximity sensors (80 sensors in total) on a robot arm, and tactile information is fed into a planner such that the robot can avoid obstacles in real time. Flexible, bendable, and stretchable sensor matrices with pressure and temperature sensors used as a conformable skin for robots have been introduced in [7]. The manufacturing of such a skin has been shown to be feasible up to a size of 12×12. Using flexible sensor matrices as a solution for robot skin has also been investigated in [8]. However, sensor matrices for the robot skin have the disadvantage of high wire counts and low robustness since matrices are susceptibility to row-/column-wise failures. The work of [9] improves the deployability of flexible tactile sensors by using a tree-like scanning topology in order to realize a cuttable sensor sheet that can be adapted to the desired shapes. To increase robustness, decrease the wire count, and improve deployability, modular approaches for the robot skin are promising. Modular approaches to robot skin systems have been investigated in [10], where the RI-MAN robot is equipped with five tactile sensing modules (two per arm and one on the chest) with an 8×8 force sensor matrix, in total 320 force sensors. Similarly, the ARMAR-III robot [11] uses the modules with force sensor matrices on its shoulders and arms. The TWENDY-ONE robot [12] is completely covered with the robot skin with force sensors distributed on its hands (2×241 sensors), on its arms (2×54 sensors), and on its trunk (26 sensors). Another example of the robot skin with modular and hierarchical architecture and strict real-time communication is presented in [9]. This robot skin was implemented to cover the whole body of a humanoid robot with 1864 sensing points. The authors further tested this system for whole-body contact tasks [13], [14]. A truly modular approach has been introduced with RoboSkin [15]. This robot skin system employs triangular skin modules with 12 capacitive force sensors. These skin modules can be assembled to larger structures that can cover arbitrary surfaces. This allows great deployment flexibility, reduction of wires, and the containment of failures to modules, paving the way for covering large areas of robots with good spatial resolution. RoboSkin has been used to cover a Nao robot with 200 force sensors and an iCub with 2000 force sensors [16], [17]. More recently, there has been progress in upgrading RobotSkin with 3-D magnetic force sensors [18], [19]. A useful large-scale deployment of the robot skin needs to address the efficient transmission, organization, and representation of tactile information. A query-based modular readout system is used for RoboSkin on the iCub [20]. This system is combined with a real-time middleware [21] to realize an efficient query-based organization and representation of tactile information. Recently, neuromorphic principles have been pursued to reduce redundancy at sensor level such that only novel information represented by events is transduced, transmitted, and processed rather than every single sample. Such event-driven systems can greatly reduce the requirements on network traffic and computational power. An event-driven force sensor has been introduced in [22], and an event-driven communication system for tactile information in humanoid robots has been introduced in [23]. First, steps toward an event-driven middleware for standard PC systems are introduced in [24]. Nevertheless, strict real-time control requires a constant control cycle time, while event-driven communication and processing create a dynamic system load variance, influencing the process scheduling of the operating system and, thus, causing jitter in the control cycle time. The disruption of real-time scheduling can be mitigated in multicore systems. In these systems, one thread may be locked for real-time control, while the other threads handle the event-driven system. Event decoders can provide a synchronous interface for accessing data that are updated on the arrival of new events to supply the synchronous clock-driven control algorithm with event-driven information.

2) Skin Control: The idea of tactile sensing and its potential capabilities for robot control has been considered for decades. With this purpose, different technologies and methods have been presented to detect contacts on a robot's body (not only the end-effectors). For example, the tactile sensor in [25] is used to detect physical contacts and compute body reactions. Furthermore, the detection of pre-contact has also been considered to prevent collisions between the robots and the environment [26], [27]. These physical interactions were enabled by including torque sensors to measure the interaction forces [28], [29].

With the development of smaller size and higher resolution tactile sensors, applications for sensitive grasping and manipulation have been presented [30]. For example, the sensitive gripper in [31] is used to find a grip without any information of the grasped object and perform stable lifting even for non-stiff objects. Sagisaka *et al.* [32], [33] presented a high-resolution tactile skin with 1052 sensing points deployed on one hand. This skin was primarily used to measure human manual skills. Tactile feedback can improve the grasping methods even for two-fingered grippers as in [34].

Skin technologies have also been used for dynamic parameter identification using learning techniques as in [35]. In the field of legged robotics, tactile sensors have been used to identify terrain conditions [36], [37] and to enable compliant walking strategies [38] and balance controllers [39], [40].

B. Outline of the Article

In the following, we present all stages of the development of our robot skin for robots. Section II presents a multimodal robot skin cell and how they are networked to form surfaces on robots. Section III presents the data handling scheme needed to support the handling of the large data coming from a large surface of the robot skin. The control framework is presented in Section IV. In Section V, examples of applications of our robot skin on various robotic use cases are illustrated. Finally, a conclusion is given in Section VI.

II. MULTIMODAL ROBOT SKIN

A. Skin Cells

The development of our robot skin started in 2010 [41] and evolved into a newer generation [42]. For the multimodal skin presented in this section, we developed an artificial multimodal robot skin cell that can sense: 1) temperatures; 2) pressure (normal forces); 3) accelerations (translation direction in: x, y, and z); and 4) proximity [see Fig. 2(a)]. Each skin cell is equipped with a microcontroller for local computation and reading of the sensors [see Fig. 2(b)], as well as providing communication between its neighbors (with four communication ports).

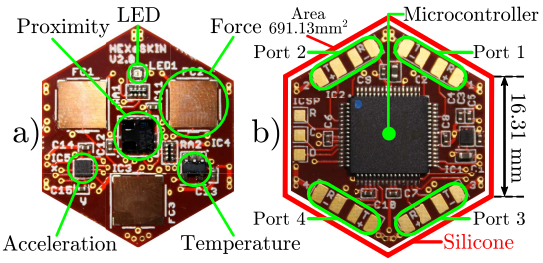


Fig. 2. ***Robot skin developed at Institute for Cognitive Systems. (a) Sensors mounted on every cell. (b) Microcontroller and dimensions of the cell electronics encapsulated in silicone material.***

B. Forming a Skin Network

The skin cells of our robot skin system connect to up to four neighboring skin cells via four communication ports [see Fig. 2(b)]. These directly connected skin cells form one entity that we call robot skin patch. These skin patches can be connected to each other by simply connecting with some of their bordering skin cells. Eventually, skin patches are connected to tactile section units (TSUs) that provide a communication interface between high-speed Gigabit Ethernet connections to the host PC and the skin network (see Fig. 3). The skin network is fully modular and self-organizing such that the only constraints for forming skin networks are power and communication bandwidth [41]. One skin cell port can support connections to up to 66 skin cells (4 Mb/s = 66 skin cells at 250 Hz).

C. Automatic Configuration and Localization of Skin Cells

To fully exploit the advantages of the robot skin in applications, the sensory information needs to be combined with spatial information, e.g., we need to label and localize the skin sensors on the robot. The skin configuration and localization can be automated, for example, combining self-touch with RGB-D information [43]. In our case, we developed a self-configuration/self-localization method. The main steps of this method are shown in Fig. 4. These steps are: 1) mounting the skin patches on the robot limbs; 2) exploring the skin sensor network to extract

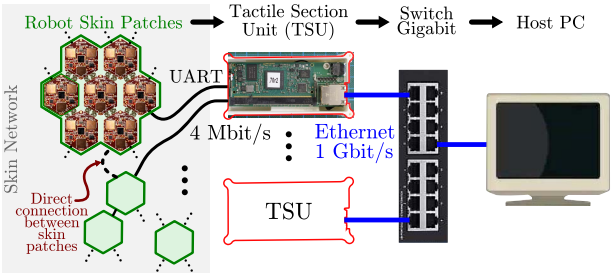


Fig. 3. ***Robot skin network and communication interface to a workstation.***

its topology; 3) motor babbling to obtain skin cells' data from different poses; 4) 3-D surface reconstruction; and 5) skin patch localization with respect to the robot limb where the patch is mounted.

To find the transformation between each skin cell i and a robot link j, we developed a method that combines net-linear acceleration information with neighbor and shape information to automatically reconstruct the 3-D surface of skin patches [44]. The advantage of this method is that it scales well since the algorithm works limb-wise and not sensor-wise, i.e., the complete sensors allocated in a robot limb (including sub-limbs, e.g., arm and forearm) are localized in a single round. This method infers the relative transformation ${}^{i}\mathbf{T}_{r}$ for each skin cell i of a patch with respect to a root skin cell r [see Fig. 4(d)]. Then, we choose one of our methods shown in Fig. 5 to localize the patch on a robot limb, i.e., we determine the transformation ${}^{r}\mathbf{T}_{j}$ between the root skin cell r of a patch and its robot limb j [see Fig. 4(e)].

The 3-D surface reconstruction method first uses the local neighbor information (for each skin cell a list of its four neighbors) and creates a directed graph. The neighbor lists are automatically acquired during the start-up process of the skin [see Fig. 4(b)]. In this directed graph, skin cells are represented by vertices and connections between the neighboring skin cells by edges. Then, the method performs a connected component analysis that separates the graph into subgraphs. Each subgraph represents a skin patch. In the next step, we determine the root cell for each skin patch. Therefore, we perform Dijkstra's shortest path

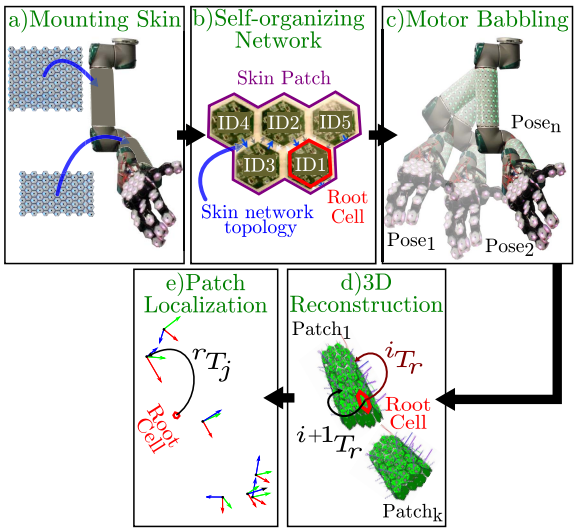


Fig. 4. *Self-configuration and self-localization of the robot skin system. Once the skin patches are mounted on the robot, the approach starts with a skin cell network exploration, then a 3-D reconstruction of the robot surface is performed, and finally, a skin patch localization is obtained. (a) Mounting skin. (b) Self-organizing network. (c) Motor babbling. (d) 3-D reconstruction. (e) Patch localization.*

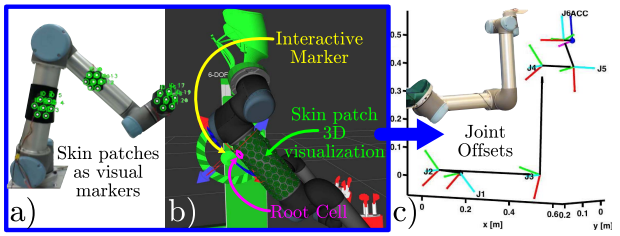


Fig. 5. *(a) Localization of robot skin patches (root cells) in the robot body frames (${}^{r}T_{j}$) using (a) embedded LEDs in the skin cells to transform the skin patches into visual markers. (b) Manual localization of the skin patches using an intuitive GUI with interactive markers. (c) D-H-like kinematic parameters can be obtained using the skin information and the local transformations from the skin patches to the robot body frames.*

first algorithm for each skin cell of a patch and calculate the accumulated path to all other skin cells. The skin cell that has the smallest accumulated path to all other skin cells in a patch is selected as a root skin cell. In this manner, we reduce the number of relative transformations ${}^{l}\mathbf{T}_{l-1}$ between the skin cells to determine ${}^{i}\mathbf{T}_{r}$ that reduces its overall error. We perform a motor babbling and measure the gravitational acceleration observed by each skin cell in different poses [see Fig. 4(c)]. Then, we compute the relative transformations ${}^{l}\mathbf{T}_{l-1}$

$$ {}^{l}\mathbf{T}_{l-1} = \begin{bmatrix} {}^{l}\mathbf{R}_{l-1} & {}^{l}\mathbf{t}_{l-1} \\ \mathbf{0}^{T} & 1 \end{bmatrix} \in \mathbb{R}^{4\times 4} \tag{1} $$

between all neighboring skin cells solving the orthogonal Procrustes problem

$$ {}^{l}\mathbf{R}_{l-1} = \arg\min_{\mathbf{\Omega}} \|\mathbf{\Omega}\,\mathbf{A} - \mathbf{B}\|_{\mathrm{F}}^{2} \quad \text{s.t. } \mathbf{\Omega}^{T}\mathbf{\Omega} = \mathbf{I} \tag{2} $$

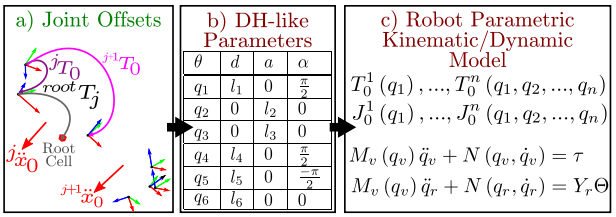
to determine the rotation ${}^{l}\mathbf{R}_{l-1}$, where $\mathbf{A} \in \mathbb{R}^{3\times n}$ contains the acceleration samples of skin cell $l-1$ and $\mathbf{B} \in \mathbb{R}^{3\times n}$ contains the samples of l, and exploit the shape information of the skin cells to determine the translation ${}^{l}\mathbf{t}_{l-1}$. Finally, we follow the shortest path from each skin cell i to the root skin cell r and calculate ${}^{i}\mathbf{T}_{r}$ using the relative transformations ${}^{l}\mathbf{T}_{l-1}$ along this path:

$$ {}^{i}\mathbf{T}_{r} = {}^{l_1}\mathbf{T}_{r} \cdot {}^{l_2}\mathbf{T}_{l_1} \cdots {}^{l_m}\mathbf{T}_{l_{m-1}} \cdot {}^{i}\mathbf{T}_{l_m}. \tag{3} $$

This automatic configuration and localization method considers the flexible connection between the hard skin cells, allowing surface conformability. Nevertheless, the 3-D reconstruction approach assumes that the relative distance between the skin cells is constant. Thus, to this point, stretchable materials are not considered.

D. Self-Acquired Kinematic/Dynamic Knowledge

The self-acquired kinematic knowledge in this article is based on our kinematic tree algorithm explained



θ	d	a	α
q_1	l_1	0	$\frac{\pi}{2}$
q_2	0	l_2	0
q_3	0	l_3	0
q_4	l_4	0	$\frac{\pi}{2}$
q_5	l_5	0	$\frac{-\pi}{2}$
q_6	l_6	0	0

Fig. 6. ***Parametric kinematic and dynamic modeling. (a) Self-acquired kinematic information of the skin patches can be used to obtain robot kinematic parameters. These parameters can be represented as (b) D-H like parameters, which in turn, can be used to obtain (c) analytic models of the robot.***

in [45]. Here, we provide a summary of the overall concept. Our approach is based on two methods. The open-loop method generates an activity matrix (AM) [45] mapping tactile stimuli with joint motion. This method can be considered as local kinematic modeling (non-geometric/analytic form association). The second method is divided into two stages: skin patch localization with respect to the robot body frames and the estimation of the joint offsets using the skin information to obtain D-H-like parameters, describing the robot kinematic configuration [46]. The skin patch localization can be obtained automatically [47] or manually using an interactive graphical user interface (GUI) (see Fig. 5).

The automatic/manual localization of the skin cells can be used to extract kinematic parameters of the robot. In this case, we use the multimodal sensor information of each cell to determine the axis of motion of the joints and its associated offsets (radii). This calibration is obtained in three phases as follows.

1) Estimating the joint axis vector based on the IMU information of each cell, while the robot generates joint-wise motions.
2) Extracting the tangential component of the acceleration during the joint motion. In this step, we subtract the gravity vector from the measurements and compute the tangential acceleration using singular value decomposition.
3) Computing the radial minimum distance, using the magnitude of the tangential component and fitting a least-squares linear model [46].

The obtained parameters can be described in the form of DH-parameters (see Fig. 6). Furthermore, combining the kinematic models obtained from the DH-parameters and the relative transformations generated during the automatic localization of the skin cells (see Section II-C), we can automatically generate the relative and absolute homogeneous transformations of each skin cell with respect to the robot's base frame, i.e., ${}^{i}\mathbf{T}_0$ and ${}^{i}\mathbf{T}_j$, where i is the ith skin cell, j is the jth robot joint, and 0 is the robot base frame (see Fig. 10). The DH-parameters and homogeneous transformations are an important step to transform the skin signals into control signals, as described in Section IV.

E. Mounting and Covering Robot Surfaces With Robot Skin

1) Mounting Robot Skin on a Robot: Mounting a large number of robot skin cells on a robot is a challenging task, since the resulting system must be reliable, simple to deploy, easy to maintain and, above all, interfere as little as possible with the robot movements. These constraints were taken into account from the early stages of design. To illustrate how we addressed some key challenges, we present the details of mounting skin on one particular robot, the industrial robot, UR5.

The first design consideration was that the robot skin should be easily mounted and removed from the robot. Therefore, its fixation mechanism had to be as simple and versatile as possible. With these in mind, it was decided to use a system of fixing studs [see Fig. 7(a)], connected to each other with a set of elastics [see Fig. 7(b)], thereby allowing the mechanical strain applied to the patch to be adjusted in a simple and intuitive manner. Since skin cells are made of rigid elements, it is not easy to reliably cover surfaces with a small radius of curvature. Moreover, as the robot skin cells are encapsulated in an elastomer material, twisting a patch to fit a curved surface lengthens the outer cover, which, as a result, generates parasitic stresses on the force transducers. We solved these issues by placing a layer

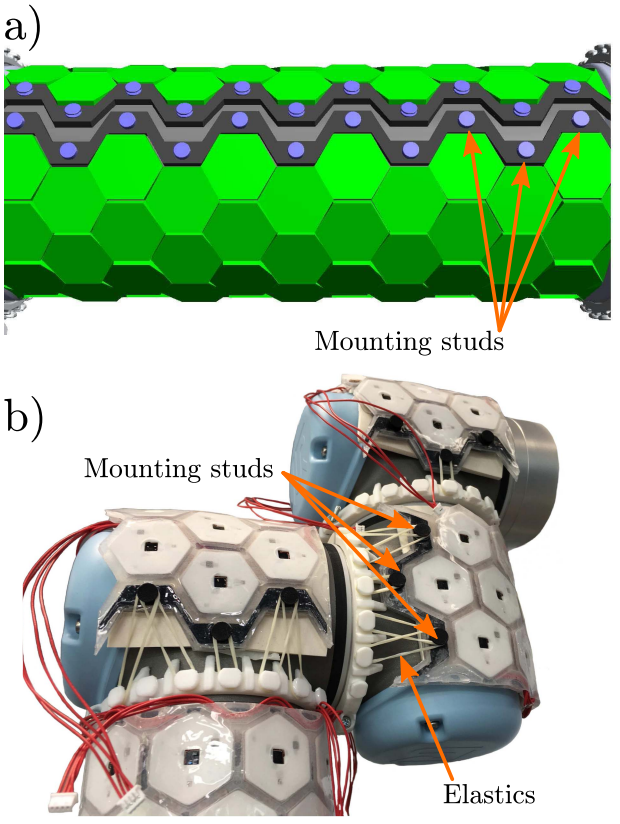


Fig. 7. ***Anchor mechanism for the robot skin. The mounting studs allow fast deployment of the skin patches on to the robot. (a) CAD design. (b) On the real robot.***

of soft-elastic material between the skin and the surface of the robot in order to soften sharp edges and release, to some extent, the mechanical stresses applied to the skin.

2) Cable Management System: One of the main issues when covering a robot with a large number of robot skin cells is the wiring of the cell network. Although the option of wireless communication was initially considered, it was soon discarded: due to bandwidth and latency issues and due to power distribution considerations (in this context, each skin patch has to be equipped with its own independent energy source). Under normal circumstances, the cables carry the energy and control signals of a robot pass through a set of dedicated joint hubs. This minimizes mechanical disturbances and therefore improves the reliability of the whole system. However, in many cases, these cable ducts are not accessible or just not suitable for additional wiring. Therefore, the only alternative is to circulate the remaining cables outside the robot. Although this is common practice in the industry, especially between a robot arm and a unique tool fixed on its end-effector, this soon becomes an issue as the number of connected devices or the complexity of the task to be achieved by the robot increases. Furthermore, in the case of a robot skin, the cables must not interfere with the measurement devices, and in particular, the proximity sensors mounted on the cells. Therefore, they must circulate under the cellular network, which considerably increases the risk of parasitic mechanical interactions with the robot. In practice, we managed to solve these issues using a system of cable re-winders and pulleys. The cable circulates through a set of channels whose ends are equipped with ball bearing mounted hinges in order to minimize friction. A set of cable reels (typically one per joint) maintain the cable under a constant—adjustable—tension, thereby ensuring that there is no undesired interaction with the skin proximity sensors. The entire system, adapted to a UR5 robot, is shown in Fig. 8.

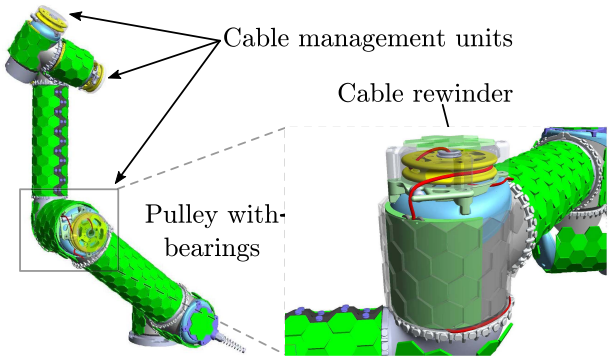


Fig. 8. *Cable management system.*

III. LARGE DATA HANDLING OF ROBOT SKIN

A. Event-Driven Artificial Skin

This section presents the data handling scheme needed to support the handling of large data coming from a large surface of the robot skin.

The key concept for efficiently handling large data of large-scale robot skin with a huge number of skin cells and sensors is to shift the extraction of meaningful information from a centralized processing system to the sensors, ideally into the sensors themselves. Such a concept implements redundancy reduction at the sensor level, which vastly reduces the amount of information to transfer to and process at the central processing system. The potential to reduce the temporal redundancy in the multimodal sensor signals of the robot skin is tremendous, as most of the time tactile sensors are only stimulated in certain areas for a short amount of time, especially in cases where robots assume upright positions and touch objects occasionally. The concept of realizing redundancy reduction at the sensor level is biologically inspired. Nature employs different kinds of specialized skin receptors that are tuned to sense elementary contact features of complex tactile stimuli and encode sensor information to neural codes represented by binary spike trains (events) in massively parallel nerve bundles [48], [49]. Most importantly, skin receptors produce spike trains only when they detected novel information. More generally, neural systems are only active when there is novel information represented by events. The application of these biological concepts results in biologically inspired systems that realize redundancy reduction by creating events only on the detection of novel information at the sensor level, encode/represent and transmit information employing events, and become active and process information on the arrival of new information. These systems are termed event-driven systems as events drive the transduction, transmission, and processing of information rather than the sampling clock as in the state-of-the-art discrete-time systems (clock-driven systems) that follow the Nyquist theorem. Fig. 9 shows our overall concept of an efficient data handling scheme based on the event-driven robot skin [50]–[52].

1) Ideal Event-Driven Sensor: The ideal event-driven sensor generates events whenever it is stimulated with novel stimuli. Such a sensor is not sampled, and any event

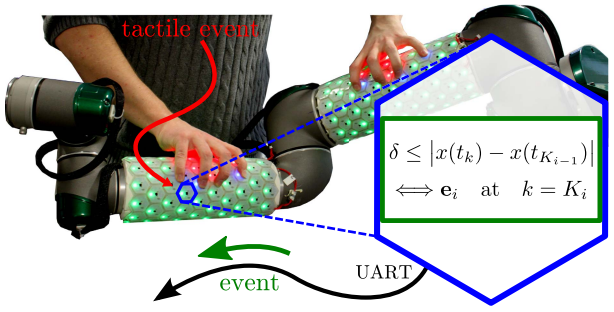


Fig. 9. *Event-driven tactile skin. Sensory signals are only forwarded to the interface box when a tactile event is generated.*

generation is completely asynchronous and can occur continuously at any time. Consequently, an ideal event-driven sensor approximates an infinite sampling rate and has a very high temporal resolution. The development of such event-driven sensors is an ongoing process that started with event-driven vision [53], [54] sensors and continued with auditory [55], and force [22] sensors. All these sensors come very close to the ideal event-driven sensor and employ approximately the same signal transduction mechanism. They integrate the derivative $\dot{x}(t)$ of an analog input signal $x(t)$ and generate an event $\mathbf{e}_i$ at time t_i whenever the integral passes a predefined threshold δ

$$\delta \leq \Big| \int_{t_{i-1}}^{t} \dot{x}(t)\, \mathrm{d}t \Big| \Longleftrightarrow \mathbf{e}_i \text{ at } t = t_i. \tag{4}$$

This integral arithmetically averages the change of the input signal $\dot{x}(t)$ such that the event-generation rule can be written as

$$\delta \leq |x(t) - x(t_{i-1})|. \tag{5}$$

Consequently, this event-generation principle realizes change-event generators that generate events whenever a monitored sensor signal changes beyond a specified limit. Besides the direct analog to event conversion, event-driven sensors employ address event representation (AER) to represent and transmit events with high temporal precision [56], [57]. The AER concept employs an asynchronous address bus that encodes the creation time of events by their occurrence on the bus and their origin (e.g., pixel address) and type by their address on the bus. Whenever events in AER need to be processed by synchronous systems such as a PC, then an asynchronous-to-synchronous bridge has to add high-precision time stamps to the events. The main advantage of AER is its high temporal precision. However, the AER requires fast asynchronous parallel bus systems. These bus systems are readily available in VLSI ICs but are hard to realize in distributed systems with hard constraints on wiring, such as robot skin systems. Recently, serial AER has been developed, which reduces the number of wires to one differential pair [58]. However, these wires still need to support very high transmission rates. Since transmitting only change events induces drift at the reconstruction of absolute values whenever events are lost, more recent event-driven sensors also encode absolute values into events using rate codes [22] or time codes [59].

2) Send-on-Delta Principle: The send-on-delta principle (SoDP) has been first proposed in [60] and [61] as an efficient concept to reduce the number of transmissions in wireless, battery powered, widely distributed sensor networks. In this setup, reducing the number of transmissions is essential to increase the life expectancy of the sensors. The SoDP realizes the following concept. To implement redundancy reduction at the sensor level, off-the-shelf sensors are sampled as fast as possible, while sensor values $x(t_k)$ with $t_k = k\,T_\mathrm{s}$, $k \in \mathbb{Z}$ are only sent at time instances $t_{k=K_i}$ when the difference between the last sent value $x(t_{K_{i-1}})$ and the currently sampled value $x(t_k)$ exceeds a predefined fixed threshold δ

$$\delta \leq |x(t_k) - x(t_{K_{i-1}})| \Longleftrightarrow \mathbf{e}_i \text{ at } k = K_i. \tag{6}$$

The similarity to the previously discussed change-event generators that directly convert analog values into change events in the continuous time domain is obvious. The event-generation rule for the SoDP is basically just the time-discrete counterpart for the time continuous rule presented in (5). The big advantage of the SoDP is that it works in compound architectures [61] with standard sensors, standard asynchronous communication protocols (e.g., UART, Ethernet), and normal time-discrete processing systems. An event $\mathbf{e}_i$ in SoDP

$$\mathbf{e}_i = \begin{pmatrix} x(t_k) & t_{K_i} \end{pmatrix} \text{ at } k = K_i \tag{7}$$

is represented by an event packet, which contains the sensor value $x(t_{K_i})$ that triggered the event and is sent on the occurrence t_{K_i} of the event (see Algorithm 1). Thus, the conversion of asynchronous event packets back to a time-synchronous stream of information for synchronous processing is drift-free and does not suffer from event losses. However, an event-driven system using SoDP and event packets has a lower temporal precision than event-driven systems using analog event generators and AER. The temporal precision of SoDP is limited by the sampling rate of the sensors and the temporal precision of the transport layer of the event packets.

3) Event-Driven Signaling for Robot Skin: We propose a simple event-driven signaling concept for robot skin systems that can be realized without special hardware or a special skin cell network architecture. This excludes using the AER principle, and we aim for a system that makes use of the SoDP. Actually, implementing the SoDP in robot skin systems exhibits two major advantages as follows.

1) By designing the robot skin system into modularized skin cells, which provides local processing capabilities at the skin cell level, with an asynchronous communication network, this robot skin system can be turned into an event-driven robot skin system without the need for special hardware.
2) The robot skin system can then provide two operation modes, the standard clock-driven mode with a constant sampling rate and an event-driven operation mode. The ability to provide two operation modes becomes useful when comparing and evaluating clock-driven applications with event-driven applications.

Algorithm 1 Event Generation

$i := 1$
$k := 0$
$\text{value}(\mathbf{e}_0) := 0$
$\text{time}(\mathbf{e}_0) := 0$
loop
  sample $x(t_k)$
  $x(t_{K_{i-1}}) = \text{value}(\mathbf{e}_{i-1})$
  if $\delta \leq \left|x(t_k) - x(t_{K_{i-1}})\right|$ **then**
    $K_i := k$
    $\text{value}(\mathbf{e}_i) := x(t_k)$
    $\text{time}(\mathbf{e}_i) := t_k$
    create event packet for $\mathbf{e}_i$
    send event packet
    $i := i + 1$
  end if
  $k := k + 1$
end loop

The robot skin system that we developed provides all the requirements for extending it with an event-driven operation mode. We implemented an event generator that creates and sends event packets following the SoDP by exploiting the microcontrollers of the skin cells. Since our self-organizing skin cell network implementation currently relies on constant packet sizes, and also to reduce the overhead generated by formatting information into packets, we pack events of different sensor modalities that occur at the same time into one single event packet. We analyzed the event generation in different applications to find the optimal packet size, such that the packets are big enough to transport several events in one packet but also small enough to reduce the overhead in cases when only one event needs to be sent [50]. We could show that in the event-driven mode, the network bandwidth reduces by up to 90%, and the CPU usage of the skin driver running at the PC reduces by up to 60% [52].

B. Dynamic Routing and Load Balancing

Skin cells in robot skin systems require a stable and robust communication network that is capable of providing reliable bi-directional connection paths between each skin cell in the network and the host PC. With larger numbers of skin cells in such networks, the task of constructing such networks becomes more challenging due to the following facts: 1) a large number of skin cells in a network increase the number of points of failure and the failure of one single skin cell or connection could affect more skin cells and larger skin regions and 2) more skin cells form larger networks with higher communication loads such that communication trees have to optimize for network depth to minimize delay and balance the load to avoid congestion. To tackle these challenges, we developed a new self-organizing network protocol for skin cells that is capable of: 1) automatic construction of bi-directional communication trees with a deterministic low network depth without prior knowledge of the network topology; 2) dynamic online re-routing of connections on the detection of broken connections or skin cells; and 3) dynamic online load balancing to fairly balance the network load between the skin cells and connections to the host. The realization of these measures in robot skin systems allows dynamic re-routing and full recovery of lost connections in only 40 ms, deterministic and reduced network depths, and improved load balancing, which reduces packet loss and delay and increases the maximum network size by up to 32% [62].

IV. ROBOT SKIN CONTROL FRAMEWORK

In this section, we describe how we integrate the multimodal tactile signals of our robot skin into control systems to enhance the reactiveness of robots in physical human–robot interaction scenarios. We use two different approaches: one is based on a robot-dependent sensory-motor map [63] and the second is a more general approach to handle different robots with distinct control interfaces [64], [65].

A. Postural Sensory-Motor Mapping

The sensory-motor map is a set of postural matrices, mapping distributed tactile stimulations into joint-level robot reactions, grounded on the reference body part. This map is similar to a Jacobian-based inverse kinematic solution for numerous distributed contact points on the surface of a robot. The map is acquired automatically and replaces manual calibration efforts for the robot kinematics, as well as the placement/orientation of skin sensors on its surface. The general idea is to mount our robot skin on the robot links, generate joint-wise motions (motor babbling), and use the correlated information of the joint states and the robot skin to generate the map. This map is generated using three main steps in the following.

1) Exploration Pattern: In order to evaluate the influence of each revolute degree of freedom (DOF) (d), on the translational motion of a sensor cell (i), in a pose (p), the robot applies test patterns to one DOF after the other. Due to the currently available tactile sensors, i.e., normal force and proximity, we focus on the translational component in the direction of the surface normal. For the translational components, only the tangential acceleration ${}^i\vec{a}_{i,d}^{tan}$ can be utilized, as it is collinear with the local motion vector. The influence of the centripetal acceleration can be minimized by keeping the angular velocity ω_d low. The influence of the rotated gravity vector is nearly constant and thus subtractable when the DOF motion only covers a small angular range $\triangle\varphi_d$. In order to maximize the tangential acceleration, the angular acceleration α_d has to be high. In order to maintain smooth accelerometer readings, it is necessary to control the angular velocity $\omega_d(t)$, the acceleration $\alpha_d(t)$, and the jerk $\zeta_d(t)$. It is desirable

that the DOF returns to its initial position $\varphi_d(0) = \varphi_d(t_f)$ once the exploration pattern stops at time t_f. One velocity control pattern $\omega_d(t)$ that fulfills all these requirements is a sine wave.

2) Pattern Evaluation: A distinct value quantifies the contribution of a DOF (d), toward the desired motion of a sensor cell (i) in the current pose (p). In order to eliminate constant sensor offsets and the gravity vector ${}^i\vec{g}$, we first subtract the mean value from all accelerometer axes. To eliminate noise and vibrations, we apply a digital low-pass filter, with a bandwidth B greater than ten times the pattern frequency f. In order to find the minimum and maximum, we calculate the amplitude for every axis along the z-axis ${}^i\vec{e}^z$ of the accelerometer $A^z_{i,d,p}$

$$A^z_{i,d,p} = \max\left({}^i a^z_{i,d}\right) - \min\left({}^i a^z_{i,d}\right). \tag{8}$$

In order to discriminate if the desired motion is in-phase or anti-phase, we evaluate if the minimum or maximum is located first in time

$$f_{\max} = \text{find}\left({}^i a^z_{i,d} == \max\left({}^i a^z_{i,d}\right), \text{'first'}\right) \tag{9}$$

$$f_{\min} = \text{find}\left({}^i a^z_{i,d} == \min\left({}^i a^z_{i,d}\right), \text{'first'}\right) \tag{10}$$

$$s^z_{i,d,p} = \text{sign}(f_{\max} - f_{\min}). \tag{11}$$

The weight $w^z_{i,d,p}$ in the local skin cell's surface normal direction ${}^i\vec{e}^z$ is now computed as

$$w^z_{i,d,p} = s^z_{i,d,p} \cdot \frac{A^z_{i,d,p}}{A^x_{i,d,p} + A^y_{i,d,p} + A^z_{i,d,p}}. \tag{12}$$

Weights have the values between $[-1; 1]$, they are close to ± 1 if the DOF motion fully correlates with the desired translational motion while being close to 0 in the orthogonal cases.

3) Sensory-Motor Map: The sensory-motor map is a container for the explored weight values, forming a lookup table for the mapping of tactile reactions. Each tile of the sensory-motor map is explored in a pose (p) and features up to three sets of matrices that are related to the three translational directions. The dimension of each matrix is defined by the available skin cells and DOFs ($U \times D$). With our current set of sensory modalities, we only make use of the matrix values $w^z_{i,d,p}$ collinear to the surface normal. Each tile also contains a vector of the robot pose that it has been explored in, which helps to recall the closest (e.g., quadratic distance) memorized pose when mapping tactile reactions into robot reactions. On robots with multiple kinematic chains, e.g., a humanoid with two arms, dynamic coupling effects in between the moving body parts are likely. Post-processing of the postural matrices with the structural knowledge helps to decouple those body parts. Here, we element-wise multiply ($\circ$) the global AM with each sensory-motor map matrix (W_p)

$$W_{p,\text{new}} = AM \circ W_p. \tag{13}$$

Additionally, we remove all small skin cell reaction vectors ($\vec{w}^z_{i,p}$), because those reactions cannot be grounded on a static reference, such as the torso. Vectors above this suppression threshold are normalized to balance the tactile reaction strength along the entire kinematic chain. The sensory map obtained with this approach has been evaluated in a humanoid robot (see Section V-C).

B. Skin General Control Framework

The control framework presented in this article was first proposed in [64] for the robot TOMM [66] with two 6-DOF industrial robot arms (UR-5), and thereafter, it has been extended to support other robots [65].

The pipeline of our general control framework is shown in Fig. 10. The framework is a multilevel hierarchical control system, whose main purpose is to generate adequate robot behaviors according to an objective defined by the user and to transform the control output using a torque resolver module, such that the correct command interface is selected according to the type of robot. The principal modules of the framework are: 1) primitive generator; 2) robot task library; 3) control fusion; 4) gain selector; and 5) torque resolver.

1) Primitive Generator: This module receives a goal from the user and generates the reference dynamics for the robot, selecting the different robot tasks, and their fusion method needed to produce the target dynamics in the robot. The definition of the reference and the tasks depends on the user-defined objective. The reference can be a combination of references in the joint, operational, and force spaces. The tasks can be any combination of tasks defined in the robot task library.

2) Robot Task Library: It contains the definitions of tasks (and their associated control approaches). All the robot tasks are defined at the generalized force level (joint torque level). We selected the torque level since it can combine tasks defined in different spaces without a complex reference frame re-allocation. The other reason to select torque is that we can distribute the total torque computation easily, exploiting the hardware embedded in the skin cells (see Section IV-B6). Examples of available tasks in the library are the Cartesian task, the joint task, both for single-limb and multilimb, and the balance task. One task that is of particular interest is the skin task. This task uses feedback information from the skin patches to produce reactive robot behaviors to tactile information. In order to integrate the information from the robot skin sensors with the other tasks available in the robot task library (see Fig. 10), we need to transform the sensor signals (e.g., proximity and normal force) to generalized force commands. We use wrench vectors to describe the

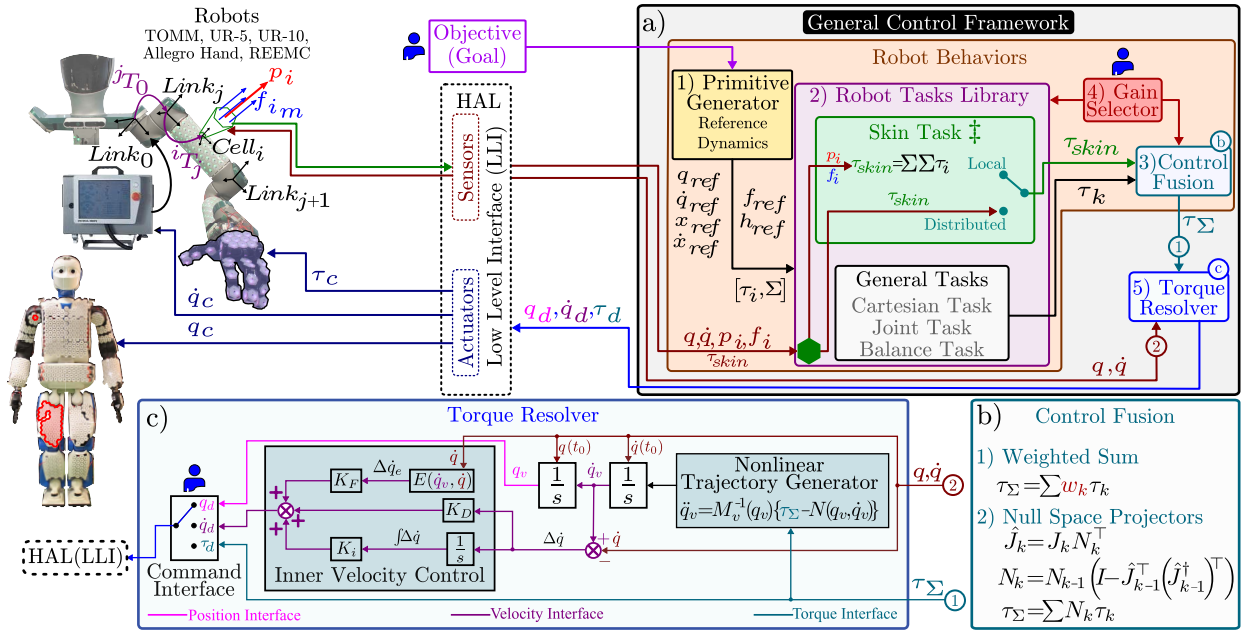


Fig. 10. ***(a) Pipeline of our general control framework. The framework is composed of five principal modules, namely, primitive generator, robot task library, control fusion, gain selector, and torque resolver. These modules produce robot behaviors that are commanded to a robot using a hardware abstraction layer (HAL). (b) Different robot behaviors are obtained by fusing multiple controllers. (c) Framework is designed to control robots with different control interfaces, i.e., joint position, velocity, and torque.***

tactile signals and map them into joint torque commands. To this aim, we perform the following transformations.

a) Multimodal tactile signals to force vector: Cell_i produces a set of three force signals $f_{i_m} \in \mathbb{R}, m = 1, 2, 3$ and a single proximity signal $p_i \in \mathbb{R}$ (see Fig. 10). By design, the force signals and the proximity signal are normal to the sensor surface, defined by its z-axis. Therefore, a virtual force vector can be calculated for each Cell_i

$$\overline{P_i} = [0, 0, w_p p_i]^T, \ \overline{F_i} = [0, 0, w_f \sum_{m=1}^{3} f_{i_m}]^T \tag{14}$$

where $w_p, w_f \in \mathbb{R}$ are weighting gains. The above equations describe the virtual force vectors of each signal with respect to the Cell_i frame (see Fig. 10). The virtual force vector generated by the proximity and force signals with respect to the robot base (Link_0) is obtained as

$${}^{i}F_0 = {}^{i}R_0(\overline{F_i} + \overline{P_i}) \tag{15}$$

where ${}^{i}F_0 \in \mathbb{R}^3$ represents a virtual force vector generated by the tactile signals of the Cell_i. The rotation matrix ${}^{i}R_0 \in \text{SO}(3)$ is extracted from the homogeneous transformation ${}^{i}T_0$, as explained in Section II-D.

b) Force vector to joint torques: In the second step, the torque $\tau_i \in \mathbb{R}^n$ produced by the tactile signals of each Cell_i is calculated as

$$\tau_i = {}^{i}J_0^T \left({}^{i}W_0\right) \in \mathbb{R}^n \tag{16}$$

where ${}^{i}W_0 = [{}^{i}F_0^T, 0^{1\times 3}]^T \in \mathbb{R}^6$ is the virtual wrench applied on Cell_i.[1] The skin joint torque $\tau_{skin} \in \mathbb{R}^n$ generated by all the skin cells on every patch k is computed as

$$\tau_{skin} = \sum_{k=1}^{p} \sum_{i=1}^{c} \tau_{k,i} \in \mathbb{R}^n \tag{17}$$

with c as the number of skin cells in a skin patch and p as the total number of skin patches on the robot. The task described by the controller of (17) can be fused with other different tasks in the robot task library to produce specific robot behaviors. The robot behaviors are defined by a specific selection of controllers with a specific control fusion method. For example, the skin-compliance behavior is composed of a joint task, a gravity compensation task, and the skin task, fused with the method defined in (18). This behavior transforms a stiff industrial robot into a fully compliant robot. For more robot behaviors, see Fig. 11.

3) Control Fusion: In order to generate an adequate robot behavior, it is important to select both the correct tasks and the method to combine (fuse) these tasks. In our general framework, we use different approaches to fuse the tasks to a single control output τ_Σ, e.g., weighted sum [64] and null-space projection [67]. In the first case, we use a

[1]We set the moment on $\text{Cell}_i = 0 \in \mathbb{R}^{3\times 1}$ since, given the neglectable distance between the force sensors and the Cell_i reference frame, it is physically impossible to apply a pure moment to an individual Cell_i with respect to its own reference frame or even measure it with the skin sensors.

Robot Behaviors	Low-Level Tasks						
	Joint Task	Spline Joint Task	Cartesian Task	Spline Cartesian Task	Skin Task	Skin Cartesian Task	G Task
Skin Compliance	✓				✓		✓
Reach Joint Goal Compliant		✓			✓		✓
Reach Cartesian Compliant			✓		✓		✓
Reach Cartesian Goal Compliant				✓		✓	✓
Kinesthetic Joint					✓		✓
Kinesthetic Cartesian						✓	✓

Fig. 11. ***Robot behaviors and their associated tasks.***

simple fusion method which is defined as

$$\tau_\Sigma = \sum_{k=1}^{m} w_k \tau_k \in \mathbb{R}^n \tag{18}$$

where w_k is a weight gain to control the contribution of each control law.

This is a simple and direct method to fuse the control output of different tasks, which can generate complex behaviors such as a compliant behavior in a noncompliant robot [64], where we fused the joint with the gravity compensation and the skin tasks. The second method that we implemented in our general framework is based on a stack of tasks. A strict task-hierarchy execution policy can be imposed over a set of tasks using null-space projectors, as described in [68]. For a set of m tasks, the whole body joint torque with hierarchical task execution is defined as

$$\tau_\Sigma = \sum_{k=1}^{m} N_k \tau_k \in \mathbb{R}^n \tag{19}$$

with the null-space projectors N_k defined with a recursive algorithm as

$$\begin{aligned} N_1 &= I \in \mathbb{R}^{n\times n} \\ \hat{J}_k &= J_k N_k^\top \in \mathbb{R}^{s\times n} \\ N_k &= N_{k-1}\left(I - \hat{J}_{k-1}^\top \left(\hat{J}_{k-1}^\dagger\right)^\top\right) \end{aligned} \tag{20}$$

where $\hat{J}_k$ is the augmented Jacobian matrix of the kth task and the superscript † is the generalized pseudoinverse of a matrix. s is the task space dimension.

4) Gain Selector: Another important factor that contributes to the total robot behavior is the proper selection of control gains. This module selects an adequate set of gains based on the objective. These gains are tuned and stored during the calibration of the controllers.

5) Torque Resolver: The robot control framework shown in Fig. 10 aims to provide control commands in a general form for different robot architectures. It is designed to provide the three common low-level control interfaces available in standard robots, i.e., position, velocity, or torque interfaces, where the first two are the most common interfaces used in most of the modern industrial robots. In the case of torque interfaced robots, we use directly the fused control signal τ_Σ as a control command. In order to control robots with position or velocity interfaces, we need to transform τ_Σ into desired joint positions/velocities, respectively. We have implemented a torque resolver [see Fig. 10(c)], which uses the dynamic state of a nonlinear observer to generate the desired joint commands. We obtain the full dynamic model to design the observer using the kinematic models of the robot in combination with the parametric dynamic model explained in Section II-D. This parametric model allows to specify user-defined dynamic behaviors, e.g., it can increase the viscous friction, thus generating a slower step response to the tactile interaction. The desired joint positions/velocities $(q_d, \dot{q}_d)$ generated by the torque resolver are sent to the robot using its standard control interface. The torque resolver is defined by two principal modules: 1) the nonlinear trajectory generator that produces the desired trajectories based on user-defined dynamic behaviors and 2) the inner velocity control that generates a desired joint velocity in order to compensate uncertainties in the robot parameters, e.g., dynamic friction. $q_v, \dot{q}_v$ represent the joint position/velocity of a virtual robot (target position/velocity), $q, \dot{q}$ are the joint position/velocity of the real robot, and $q_c, \dot{q}_c$ are the commanded joint position/velocity sent to the robot using its control interface. $E(\dot{q}_v, \dot{q})$ is a joint velocity estimator.

This general control framework has been successfully validated in multiple robots under different conditions. Examples of these applications are elaborated in Section V.

6) Distributed Computation of Skin Joint Control: The skin task computation presented in Section IV-B2b is the key element of the skin task in the robot task library and is thus part of all robot behaviors that consider tactile inputs [see Fig. 10(b)]. However, computing the joint torque contribution $\boldsymbol{\tau}_i$ of a skin cell i involves computing a geometric Jacobian $\mathbf{J}_i^T(\mathbf{q})$ and solving (16) for each skin cell. Computing the skin joint torque contributions in the skin task for thousands of skin cells in the control loop is computationally expensive and challenging. In order to tackle this challenge and to relax the limit of the maximum number of skin cells that can be used in control, we exploit the processing capabilities of the microcontrollers of the skin cells and distribute the centralized skin joint torque computation from the PC to the skin cells [69]. This decentralization requires the realization of the following steps in each skin cell.

1) Store the static transformation ${}^i\mathbf{T}_l$ between skin cell i and robot link j.
2) Receive the most recent joint states $\mathbf{q}$.
3) Compute the forward kinematics to the skin cell i on robot link j, and thus, all ${}^l\mathbf{T}_0(\mathbf{q})$ for $l \leq j$.
4) Compute the Jacobian $\mathbf{J}_i^T(\mathbf{q})$ of skin cell i.
5) Compute the skin joint torque contribution $\boldsymbol{\tau}_i$ of skin cell i.

6) Add τ_i to the skin joint torques $\tau_{\text{skin},n}$ of neighbors n.
7) Send the resultant skin joint torque $\tau_{\text{skin},i}$ of skin cell i.

The realization of the decentralized skin joint torque computation requires several optimizations for computing the forward kinematics and (16) since the computational power of the skin cells' microcontroller is limited. Lacking a floating-point unit, we use fixed-point arithmetics and a fast approximation for the sine/cosine functions using a fifth-order polynomial approximation [69]. The optimized computation of the forward kinematics separately computes the rotation matrices ${}^{l}\mathbf{R}_0$ and projections

$$\mathbf{p}_{i,l-1,0} = {}^{i}\mathbf{t}_0 - {}^{l-1}\mathbf{t}_0 \tag{21}$$

that are required for computing the geometric Jacobians $\mathbf{J}_i^T(\mathbf{q})$. This reduces the required number of multiplications and additions by almost a half [69]. Since the skin joint torque contribution τ_i of skin cell i does not have a frame of reference, the skin cells can add up skin joint torque contributions, while the information passes through the skin cell network to the robot. Eventually, the robot receives only one vector of skin joint torques τ_{skin} that fully implements the desired skin task [see Fig. 10(b)]. We could demonstrate that the distributed computation of skin joint torques is feasible, the accumulated errors caused by the sine approximation and the fixed point arithmetic are neglectable, and the delay of the skin task in the control loop becomes independent to the number or skin cells [69].

V. APPLICATIONS OF ARTIFICIAL ROBOT SKIN

In this section, we wish to present some of the applications where we have applied our robot skin system. We first show the application on a dual-arm industrial robotic system [66], which was part of the European project "Factory-in-a-Day." The objectives of this project were to set up a factory within 24 h, where the robot skin played an essential role. Later, using the same robot, we implemented a safe force propagation approach [70]. Then, we implemented a reactive grasp control in a 16-DOF robot hand. This was followed by the implementations of the robot skin in two humanoid robots [65], [71]. Finally, we present the results obtained using the robot skin in the "Walk Again" project.

A. Intuitive Industrial Robot Teaching and Control Using Robot Skin

This section provides a brief description of an application of our robot skin on to industrial robot arms [72]. In this application, we integrated the self-organizing/self-calibrating robot skin and the general control framework with a semantic-based teaching by demonstration approach to enhance the flexibility, usability, and safety of the industrial robot arms. The complete integration allows nonexpert users to teach robots by physically interacting with them (see Fig. 12). The robot behaviors' module of the general framework [see Fig. 10(a)] generates useful behaviors needed during the teaching and execution phases of the experiment. Fig. 11 shows some of these robot behaviors and their associated control approaches. The target task is to teach a robot how to sort oranges, where the good oranges must be placed in a container, while the bad oranges must be put in the trash bin. This task shows the benefits of using multimodal tactile information since the oranges need to be squeezed to define their quality. The combination of the multimodal information of the robot skin, the robot behaviors, and the semantic engine automatically generates a plan that can be executed and transferred to different robots and for different objects, i.e., the robot uses the obtained knowledge from the orange sorting scenario, and applies it to sort other fruits [72].

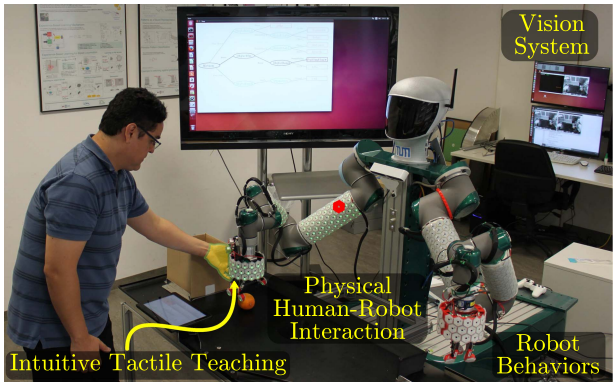


Fig. 12. *Industrial application of our skin.*

B. Skin-Based Reactive Grasping With a Four Fingers Hand

Grasping and manipulation are key skills needed for any robot. In general, we can divide the grasping approaches into two areas: approaches that rely on the object information and the ones that assume unknown objects. The former rely on accurate object information and they can only work in highly structured environments, and they are not suitable for the dynamic real world. The latter offer a more general solution, and they can be further divided into two groups, global and local grasping approaches [73]. In this application, we use a global grasping approach that assumes a simple primitive to describe the target object. Then, we enhanced this approach, using robot skin information to produce a reactive grasping behavior, which can deal with object uncertainties. The results of this application can be seen in Fig. 13. In Fig. 13(a)–(h), we present an experiment of grasping a ball that is not rigidly attached to a metal pole [see Fig. 13(a)]. The task is to grasp the

Fig. 13. *Reactive grasping using multimodal robot skin information. (a)–(d) Global grasping using simple primitives. (e)–(h) Enhanced reactive grasping using robot skin. (i)–(l) Reactive grasping for multiple objects and applying different force profiles.*

ball without dropping it. In the first case, we use the global grasping approach [see Fig. 13(a)–(d)], and due to the uncertainties in the model, the fingers collide with the ball, making it fall. In the second method [see Fig. 13(e)–(h)], we use the reactive grasping, where the fingers use the proximity sensors in the robot skin to accommodate each finger individually to the real ball surface and the force sensors to apply a homogeneous force to the ball. As a result, the ball is grasped securely and it does not fall. The same method is applied to grasp other objects with different shapes and dynamics: 1) a cylinder [see Fig. 13(i) and (j)] and 2) a wrench tool (pliers) [see Fig. 13(k) and (l)]. In the case of the wrench tool, each finger has to apply a different force profile to secure the grasping. For the experiment, we use an Allegro Hand that is a four fingers (16 DOF) hand and each DOF is torque controllable. In this case, we use our general control framework with the torque command interface.

C. Safe Whole-Body Grasping on a Humanoid Robot

In [63], we reported the whole body grasping approach that we used for a humanoid robot grasping large objects (see Fig. 14). This whole-body grasping of large unknown objects was performed on an HRP-2 robot [71]. In this setup, we used in total 74 skin cells, 12 skin cells to cover the chest, and 62 skin cells to cover the inner parts of the two arms (31 on each). We used a human-like strategy, whereby we commonly pull large and unknown objects with both arms to our chest similar to [13] and [14]. We found this approach to be effective, as three distal contact points define a stable grasp, large and compliant contact areas distribute applied forces and provide shear stability, the potentially large weight is close to the center of mass, and short kinematic chains reduce the required joint forces. In order to quickly replicate this human behavior with our artificial skin system on HRP-2, we first teach the robot grasping and pulling trajectories for both arms, then define areas (mainly on the chest and the two inner arms) to make contact with and allow forces to be applied within, employed a tactile event-driven state machine (open, close, pull, and home) to coordinate the grasping sequence, and use tactile reactions and the sensory-motor map (see Section IV-A) to adapt grasping trajectories to the size of the object.

D. Whole-Body Compliance and Interaction on a Humanoid Robot

The control methods described in IV-B can be adapted to enable compliant behaviors on systems with a high

Fig. 14. *Humanoid robot safely holding objects using the information from the robot skin and applying a sensory-motor map.*

number of DOFs, such as humanoid robots. For this purpose, the skin contact and pre-contact information can be used to define dynamic tasks following the operational space formulation [74].

The composite force (14) of the ith skin cell mounted on the Link$_j$ of the robot produces a wrench on the link. This wrench expressed with respect to the Link$_j$ reference frame is defined as

$$ {}^{i}w_j = \left[F_i^{\top}\, {}^{i}R_j^{\top}, \left({}^{i}t_j \times {}^{i}R_j F_i\right)^{\top}\right]^{\top} \in \mathbb{R}^6 \tag{22}$$

where ${}^{i}R_j$ and ${}^{i}t_j$ are the orientation and translation of the cell i in the frame Link$_j$, respectively. F_i is the composite skin force. Then, for a group of cells, the resultant wrench is

$$ W_j = \sum_{i=1}^{n_i} {}^{i}w_j \in \mathbb{R}^6 \tag{23}$$

and its projection to the robot base is defined as

$$ W_b = \begin{bmatrix} {}^{j}R_b & 0^{3\times 3} \\ 0^{3\times 3} & {}^{j}R_b \end{bmatrix} W_j \in \mathbb{R}^6. \tag{24}$$

Then, similar to the skin task (see Section IV-B), the whole-body torque vector is calculated with the Jacobian ${}^{j}J_b$

$$ \tau_s = {}^{j}J_b^{\top} W_b \in \mathbb{R}^n. \tag{25}$$

A robot with a high number of DOFs is redundant for this skin task. Therefore, other tasks can be defined to be executed in parallel, as described in [75]. Nevertheless, some tasks may have a higher priority than others. For example, all the balance-related tasks on a humanoid robot must be fulfilled for any manipulation task because the integrity of the robot and its surroundings (including human operators) depend on it.

The whole-body skin task was implemented using (19) and (20) and executed together with a balance task, a self-collision avoidance task similar to [76], and a whole body posture task. Fig. 15 shows the experimental result of two interaction cases. A multiple contact scenario in Fig. 15(a) where several skin tasks are activated at the same time and executed with a strict hierarchical policy. In Fig. 15(b), a posture task of body grasping enables the skill of a compliant hug on the robot where the size of the contact area is considerably larger (all the chest and part of the arms).

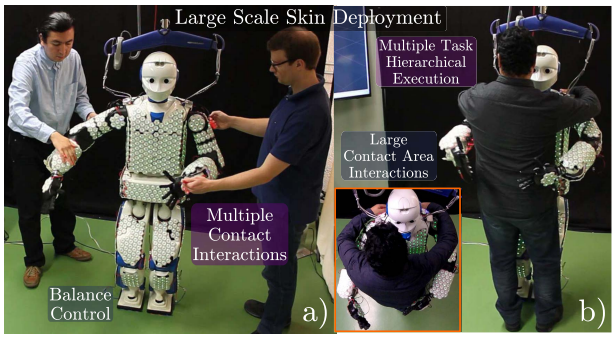


Fig. 15. *(a) Skin task can be used on the whole body for multiple contact points and executed in the null-space of the balance control. (b) Large contact area interactions are supported and feasible with the skin-compliance task definition.*

E. Force Propagation to Produce/Guarantee Whole-Body Compliance

In this section, we briefly present a control strategy that exploits the force and proximity feedback of our artificial skin to enhance the compliance range of a dual-arm mobile robot. This approach was presented in [70]. Operating in a human environment, robots must deal with a wide variety of physical interactions, likely to simultaneously occur at an arbitrary number of contact points. Whether these interactions are desired or accidental, it is important that they generate safe and predictable behaviors [78]. Therefore, such robots must be whole-body compliant, which means that compliance must be guaranteed over their entire body.

Although compliance has been widely investigated over the past two decades [79], its generalization to scenarios involving multicontact interactions between humans and stiff robots, such as industrial robots, remains challenging, on the one hand, due to the physical limitations of such robots, such as joint limits or singularities and, on the

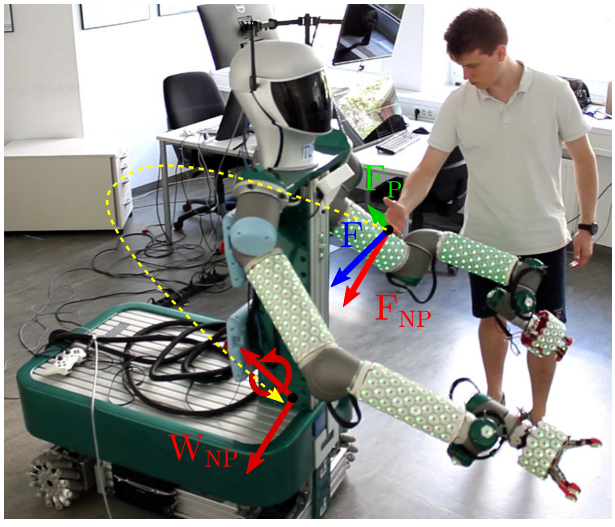


Fig. 16. *Propagation of null-space forces to produce whole-body compliance, guaranteeing safe physical human–robot interactions [70], [77].*

other hand, due to their lack of sensory inputs. In most approaches, active compliance is indeed implemented at the end-effector level [79] or at the joint level [80], resulting into potentially stiff configurations when interaction forces cannot be suitably detected or when singular configurations arise. It appears that formulating the compliance control at the contact point level allows to intuitively and precisely adjust the robot reactive behaviors at any point of its kinematic chain while providing information on null-space interactions [70], [77], [81]–[83]. In practice, covering a rigid robot with artificial skin provides it with a better awareness of its close contact environment, thereby greatly simplifying the structure of the underlying compliance control loop. In [70], the proposed controller tracks a desired—potentially safe—reactive dynamics ${}^0\ddot{\mathbf{x}}_{d_{\zeta_i}}$, defined at the contact centroid level[2] ζ_i, in Cartesian space, by the following mass-spring-damper equation:

$$ {}^0\mathbf{W}_j = \mathbf{\Lambda}_{d_j}\Delta^0\ddot{\mathbf{x}}_j + \mathbf{D}_{d_j}\Delta^0\dot{\mathbf{x}}_j + \mathbf{K}_{d_j}\Delta^0\mathbf{x}_j \tag{26} $$

where ${}^0\mathbf{W}_j \in \mathbb{R}^m$ is the external wrench applied to the contact point j and $\mathbf{\Lambda}_{d_j}, \mathbf{D}_{d_j}, \mathbf{K}_{d_j} \in \mathbb{R}^{m\times m}_+$ are the corresponding desired inertia, damping, and stiffness matrices. These matrices define the contact dynamics. $\Delta^0\mathbf{x}_j = ({}^0\mathbf{x}_{d_j} - {}^0\mathbf{x}_{ref_j}) \in \mathbb{R}^m$ and their time derivatives are, respectively, the position, velocity, and acceleration errors between the desired mass-spring-damper contact dynamics $({}^0\mathbf{x}_{d_j}, {}^0\dot{\mathbf{x}}_{d_j}$, and ${}^0\ddot{\mathbf{x}}_{d_j})$—which must be tracked by the robot—and its reference $({}^0\mathbf{x}_{ref_j}, {}^0\dot{\mathbf{x}}_{ref_j}$, and ${}^0\ddot{\mathbf{x}}_{ref_j})$, toward which the robot must converge when no force is applied to it. We formulate the compliance control law, as a set of quadratic optimization problems, solved in parallel for each limb involved in the interaction process. The objective is to find the optimal control torque $\boldsymbol{\tau}^\star$ that minimizes the acceleration residual ${}^0\ddot{\mathbf{x}}_{r_{\zeta_i}} = {}^0\ddot{\mathbf{x}}_{d_{\zeta_i}} - {}^0\ddot{\mathbf{x}}_{a_{\zeta_i}}$ while complying with a set of joint angle, velocity, and torque constraints

$$ \begin{aligned} \boldsymbol{\tau}^\star = \arg\min_{\boldsymbol{\tau}} & \sum_{i=1}^{n} {}^0\ddot{\mathbf{x}}_{r_{\zeta_i}}^\top \mathbf{\Psi}_i \, {}^0\ddot{\mathbf{x}}_{r_{\zeta_i}} + \varepsilon\boldsymbol{\tau}^\top\boldsymbol{\tau} \\ \text{s.t.} \; & -\boldsymbol{\tau}_u \le \boldsymbol{\tau}^\star \le \boldsymbol{\tau}_u \\ & \lambda(\mathbf{q}_l - \mathbf{q}) \le \boldsymbol{\tau}^\star \le \lambda(\mathbf{q}_u - \mathbf{q}) \\ & \mu(\dot{\mathbf{q}}_l - \dot{\mathbf{q}}) \le \boldsymbol{\tau}^\star \le \mu(\dot{\mathbf{q}}_u - \dot{\mathbf{q}}) \end{aligned} \tag{27} $$

where $\lambda, \mu \in \mathbb{R}$ are convergence rates and where $\mathbf{q}_l$, $\mathbf{q}_u$ $\dot{\mathbf{q}}_l$, and $\dot{\mathbf{q}}_u \in \mathbb{R}^n$ are lower and upper joint position and velocity limit vectors for the robot's upper kinematic chains. When a given limb fails to produce the desired compliant behavior, the residual dynamic at the considered contact points, mapped into a force, is propagated to a parent limb—in our case the mobile base of the robot—in order to be adequately compensated, thereby extending the whole robot compliance range (see Fig. 16).

[2]The robot skin data are merged into a single wrench vector, applied to the centroid of the considered link force distribution.

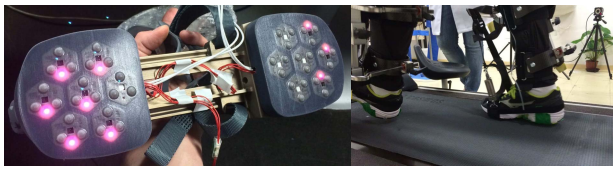


Fig. 17. *Our robot skin used on the bottom of the feet of an exoskeleton robot; on each foot 7, skin cells were used.*

F. Tactile Feedback for Rehabilitation of Spinal Cord Injured Patients

The Walk Again Project is an international project. It was initiated to study mechanisms that can enable spinal cord injured (SCI) patients to regain the function of walking. Three key technologies were investigated: 1) brain–machine interface (BMI); 2) an exoskeleton robot; and 3) vibrotactile stimulation. In this project, we applied our robot skin on the exoskeleton robot for the rehabilitation training of SCI patients. The skin cells were placed at the bottom of the feet of an exoskeleton robot (see Fig. 17), and these skin cells are used to detect touchdowns of each step. The ground contacts of the exoskeleton are translated into vibrotactile stimulation, and this is done by motors placed on the upper arm of the patient. We used three vibration motors; each vibration motor was associated with front, center, and back force sensors on the foot. The duty cycle of the pulse-width modulation signal controlling the vibration motors was set to be proportional to the measured pressure. Over time, all the patients learned to associate the feedback obtained through the force sensors as walking steps.

As reported in [84], long-term (12–14 months) training of these patients has induced the partial neurological recovery of all eight patients.

VI. CONCLUSION

This article presented a holistic solution to the development of the robot skin for robots. A step-by-step description of the developments required to deploy the robot skin into real robotic systems has been elaborated on in this article. We demonstrated the effectiveness of an artificial robot skin system that has been applied to multiple robotic platforms and various applications. Overall, we showed that the complete design of our artificial robot skin system is scalable and applicable across different robotic platforms with different control schemes (i.e., position, velocity, and torque). ■

Acknowledgments

The authors would like to thank K. Stadler and S. Stenner for their technical assistance.

REFERENCES

[1] L. D. Harmon, "Automated tactile sensing," *Int. J. Robot. Res.*, vol. 1, pp. 3–31, Jun. 1982.

[2] R. Dahiya, "E-skin: From humanoids to humans," *Proc. IEEE*, vol. 107, no. 2, pp. 247–252, Feb. 2019.

[3] R. S. Dahiya, G. Metta, M. Valle, and G. Sandini, "Tactile sensing—From humans to humanoids," *IEEE Trans. Robot.*, vol. 26, no. 1, pp. 1–20, Feb. 2010.

[4] R. S. Dahiya, P. Mittendorfer, M. Valle, G. Cheng, and V. J. Lumelsky, "Directions toward effective utilization of tactile skin: A review," *IEEE Sensors J.*, vol. 13, no. 11, pp. 4121–4138, Nov. 2013.

[5] E. Cheung and V. J. Lumelsky, "Proximity sensing in robot manipulator motion planning: System and implementation issues," *IEEE Trans. Robot. Autom.*, vol. 5, no. 6, pp. 740–751, Dec. 1989.

[6] V. J. Lumelsky, M. S. Shur, and S. Wagner, "Sensitive skin," *IEEE Sensors J.*, vol. 1, no. 1, pp. 41–51, Jun. 2001.

[7] T. Someya *et al.* "Conformable, flexible, large-area networks of pressure and thermal sensors with organic transistor active matrixes," *Proc. Nat. Acad. Sci. USA*, vol. 102, no. 35, pp. 12321–12325, 2005.

[8] M. Strohmayr, "Artificial skin for robots," Ph.D. dissertation, Fac. Informaik, Fakultät für Informatik des Karlsruher Instituts für Technologie, Karlsruhe, Germany, 2012.

[9] Y. Ohmura, Y. Kuniyoshi, and A. Nagakubo, "Conformable and scalable tactile sensor skin for curved surfaces," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2006, pp. 1348–1353.

[10] T. Mukai, M. Onishi, T. Odashima, S. Hirano, and Z. Luo, "Development of the tactile sensor system of a human-interactive robot 'RI-MAN,'" *IEEE Trans. Robot.*, vol. 24, no. 2, pp. 505–512, Apr. 2008.

[11] T. Asfour *et al.*, "ARMAR-III: An integrated humanoid platform for sensory-motor control," in *Proc. IEEE-RAS Int. Conf. Humanoid Robots*, Dec. 2006, pp. 169–175.

[12] H. Iwata and S. Sugano, "Design of human symbiotic robot TWENDY-ONE," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2009, pp. 580–586.

[13] Y. Ohmura and Y. Kuniyoshi, "Humanoid robot which can lift a 30 kg box by whole body contact and tactile feedback," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2007, pp. 1136–1141.

[14] Y. Kuniyoshi, Y. Ohmura, and A. Nagakubo, "Whole body haptics for augmented humanoid task capabilities," in *Robotics Research*, M. Kaneko and Y. Nakamura, Eds. Berlin, Germany: Springer, 2011, pp. 61–73.

[15] G. Cannata, M. Maggiali, G. Metta, and G. Sandini, "An embedded artificial skin for humanoid robots," in *Proc. IEEE Int. Conf. Multisensor Fusion Integr. Intell. Syst.*, Aug. 2008, pp. 434–438.

[16] A. Schmitz, P. Maiolino, M. Maggiali, L. Natale, G. Cannata, and G. Metta, "Methods and technologies for the implementation of large-scale robot tactile sensors," *IEEE Trans. Robot.*, vol. 27, no. 3, pp. 389–400, Jun. 2011.

[17] P. Maiolino, M. Maggiali, G. Cannata, G. Metta, and L. Natale, "A flexible and robust large scale capacitive tactile system for robots," *IEEE Sensors J.*, vol. 13, no. 10, pp. 3910–3917, Oct. 2013.

[18] T. P. Tomo *et al.*, "A new silicone structure for uSkin—A soft, distributed, digital 3-axis skin sensor and its integration on the humanoid robot iCub," *IEEE Robot. Autom. Lett.*, vol. 3, no. 3, pp. 2584–2591, Jul. 2018.

[19] A. C. Holgado *et al.*, "An adjustable force sensitive sensor with an electromagnet for a soft, distributed, digital 3-axis skin sensor," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 2582–2588.

[20] E. Baglini, S. Youssefi, F. Mastrogiovanni, and G. Cannata, "A real-time distributed architecture for large-scale tactile sensing," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2014, pp. 1663–1669.

[21] S. Youssefi, S. Denei, F. Mastrogiovanni, and G. Cannata, "A real-time data acquisition and processing framework for large-scale robot skin," *Robot. Auto. Syst.*, vol. 68, pp. 86–103, Jun. 2015.

[22] S. Caviglia, M. Valle, and C. Bartolozzi, "Asynchronous, event-driven readout of POSFET devices for tactile sensing," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, Jun. 2014, pp. 2648–2651.

[23] C. Bartolozzi *et al.*, "Event-driven encoding of off-the-shelf tactile sensors for compression and latency optimisation for robotic skin," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2017, pp. 166–173.

[24] A. Glover, V. Vasco, M. Iacono, and C. Bartolozzi, "The event-driven software library for YARP—With algorithms and iCub applications," *Frontiers Robot. AI*, vol. 4, pp. 1–7, Jan. 2018.

[25] T. Wosch and W. Feiten, "Reactive motion control for human-robot tactile interaction," in *Proc. IEEE-RAS Int. Conf. Robot. Autom. (ICRA)*, Washington, DC, USA, May 2002, pp. 3807–3812.

[26] V. J. Lumelsky and E. Cheung, "Real-time collision avoidance in teleoperated whole-sensitive robot arm manipulators," *IEEE Trans. Syst., Man, Cybern.*, vol. 23, no. 1, pp. 194–203, Jan./Feb. 1993.

[27] C. L. Boddy and J. D. Taylor, "Whole-arm reactive collision avoidance control of kinematically redundant manipulators," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 1993, pp. 382–387.

[28] G. Grunwald, G. Schreiber, A. Albu-Schäffer, and G. Hirzinger, "Programming by touch: The different way of human-robot interaction," *IEEE Trans. Ind. Electron.*, vol. 50, no. 4, pp. 659–666, Aug. 2003.

[29] D. Massa, M. Callegari, and C. Cristalli, "Manual guidance for industrial robot programming," *Ind. Robot*, vol. 42, no. 5, pp. 457–465, 2015.

[30] J. Tegin and J. Wikander, "Tactile sensing in intelligent robotic manipulation—A review," *Ind. Robot, Int. J.*, vol. 32, no. 1, pp. 64–70, 2005.

[31] E. Torres-Jara and L. Natale, "Sensitive manipulation: Manipulation through tactile feedback," *Int. J. Humanoid Robot.*, vol. 15, no. 1, p. 1850012, 2018.

[32] T. Sagisaka, Y. Ohmura, Y. Kuniyoshi, A. Nagakubo, and K. Ozaki, "High-density conformable tactile sensing glove," in *Proc. 11th IEEE-RAS Int. Conf. Humanoid Robots*, Oct. 2011, pp. 537–542.

[33] T. Sagisaka, Y. Ohmura, A. Nagakubo, K. Ozaki, and Y. Kuniyoshi, "Development and applications of high-density tactile sensing glove," in *Haptics: Perception, Devices, Mobility, and Communication*, P. Isokoski and J. Springare, Eds. Berlin, Germany: Springer, 2012, pp. 445–456.

[34] S. Chitta, M. Piccoli, and J. Sturm, "Tactile object class and internal state recognition for mobile manipulation," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2010, pp. 2342–2348.

[35] R. Calandra, S. Ivaldi, M. P. Deisenroth, E. Rueckert, and J. Peters, "Learning inverse dynamics models with contacts," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2015, pp. 3186–3191.

[36] K. Suwanratchatamanee, M. Matsumoto, and S. Hashimoto, "A simple tactile sensing foot for humanoid robot and active ground slope recognition," in *Proc. IEEE Int. Conf. Mechatronics*, Apr. 2009, pp. 1–6.

[37] Q. Yuan and J. Wang, "Design and experiment of the nao humanoid robot's plantar tactile sensor for surface classification," in *Proc. 4th Int. Conf. Inf. Sci. Control Eng. (ICISCE)*, 2017, pp. 931–935.

[38] K. Fondahl *et al.*, "An adaptive sensor foot for a bipedal and quadrupedal robot," in *Proc. 4th IEEE RAS EMBS Int. Conf. Biomed. Robot. Biomechatron. (BioRob)*, Jun. 2012, pp. 270–275.

[39] M. Shimojo, T. Araki, A. Ming, and M. Ishikawa, "A ZMP sensor for a biped robot," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2006, pp. 1200–1205.

[40] Y. Takahashi, K. Nishiwaki, S. Kagami, H. Mizoguchi, and H. Inoue, "High-speed pressure sensor grid for humanoid robot foot," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Aug. 2005, pp. 3909–3914.

[41] P. Mittendorfer and G. Cheng, "Humanoid multimodal tactile-sensing modules," *IEEE Trans. Robot.*, vol. 27, no. 3, pp. 401–410, Jun. 2011.

[42] P. Mittendorfer and G. Cheng, "Integrating discrete force cells into multi-modal artificial skin," in *Proc. 12th IEEE-RAS Int. Conf. Humanoid Robots (Humanoids)*, Nov. 2012, pp. 847–852.

[43] A. Albini, S. Denei, and G. Cannata, "Towards autonomous robotic skin spatial calibration: A framework based on vision and self-touch," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2017, pp. 153–159.

[44] P. Mittendorfer and G. Cheng, "3D surface reconstruction for robotic body parts with artificial skins," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2012, pp. 4505–4510.

[45] P. Mittendorfer and G. Cheng, "Open-loop self-calibration of articulated robots with artificial skins," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, May 2012, pp. 4539–4545.

[46] P. Mittendorfer, E. Dean, and G. Cheng, "Automatic robot kinematic modeling with a modular artificial skin," in *Proc. IEEE-RAS Int. Conf. Humanoid Robots (Humanoids)*, Madrid, Spain, Nov. 2014, pp. 749–754.

[47] P. Mittendorfer, E. Dean, and G. Cheng, "3D spatial self-organization of a modular artificial skin," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Chicago, IL, USA, Sep. 2014, pp. 3969–3974.

[48] E. P. Gardner and K. O. Johnson, *Principles of Neural Science*, 5th ed. New York, NY, USA: McGraw-Hill, 2013, ch. 23, pp. 498–529.

[49] H. P. Saal and S. J. Bensmaia, "Touch is a team effort: Interplay of submodalities in cutaneous sensibility," *Trends Neurosci.*, vol. 37, no. 12, pp. 689–697, 2014.

[50] F. Bergner, P. Mittendorfer, E. Dean-León, and G. Cheng, "Event-based signaling for reducing required data rates and processing power in a large-scale artificial robotic skin," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep./Oct. 2015, pp. 2124–2129.

[51] F. Bergner, E. Dean-León, and G. Cheng, "Event-based signaling for large-scale artificial robotic skin—Realization and performance evaluation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2016, pp. 4918–4924.

[52] F. Bergner, E. Dean-León, and G. Cheng, "Efficient event-driven reactive control for large scale robot skin," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May/Jun. 2017, pp. 394–400.

[53] P. Lichtsteiner, C. Posch, and T. Delbruck, "A 128×128 120 db 15 μs latency asynchronous temporal contrast vision sensor," *IEEE J. Solid-State Circuits*, vol. 43, no. 2, pp. 566–576, Feb. 2008.

[54] C. Posch, D. Matolin, and R. Wohlgenannt, "An asynchronous time-based image sensor," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2008, pp. 2130–2133.

[55] A. van Schaik and S.-C. Liu, "AER EAR: A matched silicon cochlea pair with address event representation interface," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2005, pp. 4213–4216.

[56] M. Mahowald, "VLSI analogs of neuronal visual processing: A synthesis of form and function,"

Ph.D. dissertation, California Inst. Technol., Pasadena, CA, USA, 1992.

[57] K. A. Boahen, "Point-to-point connectivity between neuromorphic chips using address events," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 47, no. 5, pp. 416–434, May 2000.

[58] P. M. Ros, M. Crepaldi, C. Bartolozzi, and D. Demarchi, "Asynchronous dc-free serial protocol for event-based AER systems," in *Proc. IEEE Int. Conf. Electron., Circuits, Syst. (ICECS)*, Dec. 2015, pp. 248–251.

[59] C. Posch, D. Matolin, and R. Wohlgenannt, "A QVGA 143 dB dynamic range frame-free PWM image sensor with lossless pixel-level video compression and time-domain CDS," *IEEE J. Solid-State Circuits*, vol. 46, no. 1, pp. 259–275, Jan. 2011.

[60] M. Neugebauer and K. Kabitzsch, "A new protocol for a low power sensor network," in *Proc. IEEE Int. Conf. Perform., Comput., Commun.*, Apr. 2004, pp. 393–399.

[61] M. Miskowicz, "Send-on-delta concept: An event-based data reporting strategy," *Sensors*, vol. 6, no. 1, pp. 49–63, 2006.

[62] C. Bader, F. Bergner, and G. Cheng, "A robust and efficient dynamic network protocol for a large-scale artificial robotic skin," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 1600–1605.

[63] P. Mittendorfer, E. Yoshida, and G. Cheng, "Realizing whole-body tactile interactions with a self-organizing, multi-modal artificial skin on a humanoid robot," *Adv. Robot.*, vol. 29, no. 1, pp. 51–67, 2015.

[64] E. Dean-León, F. Bergner, K. Ramirez-Amaro, and G. Cheng, "From multi-modal tactile signals to a compliant control," in *Proc. IEEE-RAS 16th Int. Conf. Humanoid Robots (Humanoids)*, Nov. 2016, pp. 892–898.

[65] E. Dean-León, J. R. Guadarrama-Olvera, F. Bergner, and G. Cheng, "Whole-body active compliance control for humanoid robots with robot skin," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 5404–5410.

[66] E. Dean-León *et al.*, "TOMM: Tactile omnidirectional mobile manipulator," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May/Jun. 2017, pp. 2441–2447.

[67] A. Dietrich, C. Ott, and A. Albu-Schäffer, "An overview of null space projections for redundant, torque-controlled robots," *Int. J. Robot. Res.*, vol. 34, no. 11, pp. 1385–1400, 2015.

[68] L. Sentis and O. Khatib, "Synthesis of whole-body behaviors through hierarchical control of behavioral primitives," *Int. J. Humanoid Robot.*, vol. 2, no. 4, pp. 505–518, 2005.

[69] F. Bergner, E. Dean-León, and G. Cheng, "Efficient distributed torque computation for large scale robot skin," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 1593–1599.

[70] Q. Leboutet, E. Dean-León, F. Bergner, and G. Cheng, "Tactile-based whole-body compliance with force propagation for mobile manipulators," *IEEE Trans. Robot.*, vol. 35, no. 2, pp. 330–342, Apr. 2019.

[71] K. Kaneko *et al.*, "Humanoid robot HRP-2," in *Proc. IEEE-RAS Int. Conf. Robot. Autom. (ICRA)*, New Orleans, LA, USA, Apr. 2004, pp. 1083–1090.

[72] E. Dean-León, K. Ramirez-Amaro, F. Bergner, I. Dianov, and G. Cheng, "Integration of robotic technologies for rapidly deployable robots," *IEEE Trans. Ind. Informat.*, vol. 14, no. 4, pp. 1691–1700, Apr. 2018.

[73] Q. Lei, J. Meijer, and M. Wisse, "A survey of unknown object grasping and our fast grasping algorithm-C shape grasping," in *Proc. 3rd Int. Conf. Control, Autom. Robot. (ICCAR)*, Apr. 2017, pp. 150–157.

[74] O. Khatib, "A unified approach for motion and force control of robot manipulators: The operational space formulation," *IEEE J. Robot. Autom.*, vol. RA-3, no. 1, pp. 43–53, Feb. 1987.

[75] C. Samson, B. Espiau, and M. L. Borgne, *Robot Control: The Task Function Approach*. London, U.K.: Oxford Univ. Press, 1991.

[76] A. Dietrich, T. Wimbock, A. Albu-Schaffer, and G. Hirzinger, "Integration of reactive, torque-based self-collision avoidance into a task hierarchy," *IEEE Trans. Robot.*, vol. 28, no. 6, pp. 1278–1293, Dec. 2012.

[77] Q. Leboutet, E. Dean-León, and G. Cheng, "Tactile-based compliance with hierarchical force propagation for omnidirectional mobile manipulators," in *Proc. IEEE-RAS 16th Int. Conf. Humanoid Robots (Humanoids)*, Nov. 2016, pp. 926–931.

[78] R. Alami *et al.*, "Safe and dependable physical human-robot interaction in anthropic domains: State of the art and challenges," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2006, pp. 1–16.

[79] N. Hogan, "Impedance control: An approach to manipulation: Part III—Applications," *J. Dyn. Syst., Meas., Control*, vol. 107, no. 1, pp. 17–24, 1985.

[80] M. Zinn, B. Roth, O. Khatib, and J. K. Salisbury, "A new actuation approach for human friendly robot design," *Int. J. Robot. Res.*, vol. 23, nos. 4–5, pp. 379–398, 2004.

[81] E. Magrini, F. Flacco, and A. De Luca, "Control of generalized contact motion and force in physical human-robot interaction," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2015, pp. 2298–2304.

[82] E. Magrini and A. De Luca, "Human-robot coexistence and contact handling with redundant robots," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Sep. 2017, pp. 4611–4617.

[83] K. S. Kim and L. Sentis, "Human body part multicontact recognition and detection methodology," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May/Jun. 2017, pp. 1908–1915.

[84] A. R. Donati *et al.*, "Long-term training with a brain-machine interface-based gait protocol induces partial neurological recovery in paraplegic patients," *Sci. Rep.*, vol. 6, p. 30383, Aug. 2016.

ABOUT THE AUTHORS

Gordon Cheng (Fellow, IEEE) received the bachelor's and master's degrees in computer science from the University of Wollongong, Wollongong, NSW, Australia, in 1991 and 1993, respectively, and the Ph.D. in systems engineering from the Department of Systems Engineering, Australian National University, Canberra, ACT, Australia, in 2001.

He has extensive industrial experience in consultancy as well as contractual development of large software systems. He was also the Founder and CEO of the company GTI Computing, Sydney, NSW, Australia, from 1995 to 2006, which specialized in networking and transport management systems in Australia. He has been Chair of Cognitive Systems, Technical University of Munich, Munich, Germany, since 2010. He is currently the Founder and the Director of Institute for Cognitive Systems, Faculty of Electrical and Computer Engineering, Technical University of Munich. He has been the Coordinator of the Center of Competence Neuro-Engineering since 2013. He has also been the Director of the Elite Master of Science Program in Neuroengineering (MSNE) of the Elite Network of Bavaria since 2016. He is also the co-inventor of approximately 20 patents and the author of approximately 300 technical publications, proceedings, editorials, and book chapters. His current research interests include humanoid robotics, cognitive systems, artificial robot skin, brain–machine interfaces, bio-mimetic of human vision, computational neuroscience of vision, action understanding, human–robot interaction, active vision, and mobile robot navigation.

Dr. Cheng was named IEEE Fellow 2017 for his contributions in humanoid robotic systems and neurorobotics.

Emmanuel Dean-Leon (Member, IEEE) received the M.Sc. and Ph.D. degrees in mechatronics from the Center for Research and Advanced Studies (CINVESTAV-IPN), Mexico City, Mexico, in 2003 and 2006, respectively.

Since 2013, he has been a Senior Researcher with the Chair for Cognitive Systems, Technical University of Munich (TUM), Munich, Germany. His current research interests include robotics, low-level control, and physical human-robot interaction/collaboration.

Florian Bergner (Student Member, IEEE) received the B.Sc. and M.Sc. degrees in electrical engineering from the Technical University of Munich (TUM), Munich, Germany, in 2011 and 2014, respectively. He is currently working toward the Ph.D. degree in electrical engineering at the Chair for Cognitive Systems, TUM.

His current research interests include multimodal tactile sensing, large-scale robot skin, sensor networks, sensor fusion, neuromorphic engineering, and event-driven signaling and processing.

Julio Rogelio Guadarrama Olvera (Student Member, IEEE) received the B.Sc. degree in mechatronic engineering from the Interdisciplinary Professional Unit in Engineering and Advanced Technologies, National Polytechnic Institute, Mexico City, Mexico, in 2011, and the M.Sc. degree in electrical engineering from the Center for Research and Advanced Studies, National Polytechnic Institute, in 2013. He is currently working toward the Ph.D. degree at the Chair for Cognitive Systems, Department of Electrical and Computer Engineering, Technical University of Munich, Munich, Germany.

His current research interests include the development of interaction and locomotion controllers for biped humanoid robots, exploiting several sensing modalities.

Quentin Leboutet (Student Member, IEEE) received the M.Eng. degree in mechatronics from the École Nationale Supérieure de l'Électronique et de ses Applications (ENSEA) Cergy-Pontoise, France, and the M.Sc. degree in electrical engineering and computer science from the Technical University of Munich, Munich, Germany, where he is currently working toward the Ph.D. degree in robotics at the Institute for Cognitive Systems.

His current research interests include robot dynamics, parameter estimation, optimal control, sensor fusion, and human–robot interaction/collaboration.

Philipp Mittendorfer (Member, IEEE) received the bachelor's degree in electrical engineering and the Diploma degree from the Technical University of Munich, Munich, Germany, in 2008 and 2009, respectively, and the Dr.Ing. degree from the Institute for Cognitive Systems, Technical University of Munich, in 2016, under the supervision of Prof. G. Cheng.

His current research interests include methods to self-organize intelligent surface sensor networks and their practical implementation as multimodal artificial sensor skins into robotic systems.

Dr. Mittendorfer received the Rohde and Schwarz Ph.D. Price and the Advanced Robotics Best Journal Paper Award in 2016 for his Dr.Ing. degree.