HAL
archives-ouvertes.fr

On Model Adaptation for Sensorimotor Control of Robots

David Navarro-Alarcon, Andrea Cherubini, Xiang Li

▶ To cite this version:

David Navarro-Alarcon, Andrea Cherubini, Xiang Li. On Model Adaptation for Sensorimotor Control of Robots. 38th Chinese Control Conference, 2019, Guangzhou, China. hal-02088055

HAL Id: hal-02088055
https://hal.archives-ouvertes.fr/hal-02088055

Submitted on 2 Apr 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

On Model Adaptation for Sensorimotor Control of Robots

David Navarro-Alarcon[1], Andrea Cherubini[2] and Xiang Li[3]

1. The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong
E-mail: dnavar@polyu.edu.hk

2. University of Montpellier / LIRMM, Montpellier, France
E-mail: andrea.cherubini@lirmm.fr

3. The Chinese University of Hong Kong, Shatin, New Territories, Hong Kong
E-mail: xli@mae.cuhk.edu.hk

Abstract: In this expository article, we address the problem of computing adaptive models that can be used for guiding the motion of robotic systems with uncertain action-to-perception relations. The formulation of the uncalibrated sensor-based control problem is first presented, then, various methods for building adaptive sensorimotor models are derived and analysed. Finally, the proposed methodology is exemplified with two cases of study.

Key Words: Sensorimotor control, robotics, adaptive systems, sensor-based control.

1 INTRODUCTION

Sensor-based control encompasses a family of methods that exploit feedback information from (typically external) sensors for controlling the robot's motion, and in general, its behaviour. On its most fundamental form, it can trace back its origins to the servomechanism problem [1]. Some common examples are visual servoing [2], tactile/force servoing [3, 4], proximity servoing [5], aural servoing [6], deformation/shape servoing [7, 8], to name a few cases.

To effectively execute these types of advanced tasks, sensor-based controls invariably require some knowledge (at least coarse) of how the robot's motor actions transform into sensor measurements. This information is captured by the *sensorimotor model* of the system, which besides coordinating action and perception, it can also be used to anticipate the effect that an input motor command will produce on the output sensor measurements [9]. Note, however, that if this information is not known (or is highly uncertain), the robot cannot properly coordinate actions with perceptions.

Existing methods to obtain sensorimotor models require either exact knowledge of its analytical structure [10, 11] (which might not be known) or only compute instantaneous local estimations of it [12] (therefore, they cannot globally describe and control the system). Compared to these computational approaches, the human brain has a remarkable degree of adaptability that allows it to learn new sensorimotor relations from birth through death and under multiple morphological and perceptual conditions (see e.g. the pioneering study [13]). Humans can easily coordinate hand motions through a mirror, position unknown tools attached to the body, and even recover (some) mobility after strokes.

Our aim in this paper is precisely to address the design of computational methods that efficiently provide sensor-guided robots with robust adaptation capabilities. For that, we first formulate the sensorimotor control problem of robots using uncertain perceptual/motor models. Next, we formulate various structure-based and structure-free methods to adaptively compute these unknown relations. Finally, the presented methodology is exemplified with two cases of study and discussions about its implementation are given.

The contribution of this work is that it presents a general methodology that can be used as a guideline or even a tutorial for researchers working on adaptive sensor-based control of robots. It can be applied to various configuration-dependent sensing modalities (e.g. vision, audio, thermal), and robotic platforms (manipulators, mobile robots, robot heads).

The rest of this manuscript is organised as follows: Sec. 2 presents the preliminaries of the problem, Sec. 3 derives the adaptive sensorimotor algorithms, Sec. 4 illustrates two cases of study, and Sec. 5 gives final conclusions.

2 PRELIMINARIES

2.1 Notation

Along this note we use very standard notation. Column vectors are denoted with bold small letters $\mathbf{m}$ and matrices with bold capital letters $\mathbf{M}$. Time evolving variables are represented as $\mathbf{m}_t$, where the subscript $*_t$ denotes the discrete time instant, or, the iteration step. Gradients of functions $b = \beta(\mathbf{m}) : \mathcal{M} \mapsto \mathcal{B}$ are denoted as $\nabla\beta(\mathbf{m}) = (\partial\beta/\partial\mathbf{m})^\intercal$.

2.2 Control Architecture

Consider a class of fully-actuated robotic systems whose configuration (modelling the end-effector pose) is denoted by the vector $\mathbf{x}_t \in \mathbb{R}^n$. In our formulation of the problem, it will be assumed that the motion of robotic system is commanded via a standard position/velocity controlled interface [14, 15] (which is typically found in the large majority of commercial robotic platforms). With position interfaces, the control commands $\mathbf{u}_t \in \mathbb{R}^n$ are given in terms of differential displacement motions as follows:

$$\mathbf{x}_{t+1} - \mathbf{x}_t = \mathbf{u}_t \tag{1}$$

All methods presented in this note are formulated using the above described position controls, yet, these can be easily transformed into its velocity control equivalent $\mathbf{v}_t \in \mathbb{R}^n$ by dividing both sides of (1) by the time step $\mathrm{d}t$ of the servo-loop as

$$(\mathbf{x}_{t+1} - \mathbf{x}_t)/\,\mathrm{d}t = \mathbf{u}_t/\,\mathrm{d}t \approx \mathbf{v}_t \tag{2}$$

This work is supported in part by the Research Grants Council (RGC) of Hong Kong under grant 14203917, and in part by The Hong Kong Polytechnic University (PolyU) under grant G-YBYT.

2.3 Configuration Dependant Feedback

To perform a sensorimotor task, the robot is equipped with a set of r sensors (not necessarily of the same modality) that continuously measure physical quantities whose values depend on the robot's configuration. This situation means that relative robot motions produce relative sensory changes. Some examples of configuration dependent signals (measured using either external or wrist-mounted sensors) are: geometric visual features, observed end-effector poses, forces applied onto a surface, proximity to an object, intensity of an audio source, temperature from a heat source, ultrasound images from probe, etc.

The feedback signal from the ith sensor is denoted by the vector $\mathbf{s}_t^i = g^i(\mathbf{x}_t) \in \mathbb{R}^{l_i}$, where the function $g^i : \mathbb{R}^n \mapsto \mathbb{R}^{l_i}$ represents the analytical sensor model that statically relates the instantaneous configuration with the feedback signal; all these measurements can be grouped into a single vector $\mathbf{s}_t = [\mathbf{s}_t^{1\intercal}, \dots, \mathbf{s}_t^{r\intercal}]^\intercal = g(\mathbf{x}_t) \in \mathbb{R}^l$. Sensorimotor controls often require to construct a vector of *meaningful* features to quantify and guide the task [2]. To this end, let us introduce the (possibly nonlinear) vectorial functional

$$\mathbf{y}_t = f(\mathbf{s}_t) = f(g(\mathbf{x}_t)) : \mathbb{R}^n \mapsto \mathbb{R}^m \tag{3}$$

for m as the number of feature coordinates (along this node, we assume that $f(g(\cdot))$ is smooth functional). There are three cases with this configuration-dependent structure: $n \geq m$ (more controls than features), $n \leq m$ (more features than controls), and $n = m$ (same number of features and controls). These cases have different properties that determine the regulation of $\mathbf{y}_t$.

2.4 Sensorimotor Control Problem

The differential expression that describes how the motor actions result in changes of feedback features is represented by the first-order model:

$$\mathbf{y}_{t+1} = \mathbf{y}_t + \mathbf{A}_t \mathbf{u}_t \tag{4}$$

for $\mathbf{A}_t = [\partial f / \partial \mathbf{s}_t][\partial g / \partial \mathbf{x}_t] \in \mathbb{R}^{m \times n}$ as the Jacobian matrix of the system (also known as the interaction matrix in the sensor servoing literature [16]), whose elements depend on the instantaneous configuration $\mathbf{x}_t$.

The sensorimotor control problem consists in coordinating the motor actions with the feedback signals such that a desired sensory behaviour is achieved. Without loss of generality, such behaviour is characterised as the set-point regulation of $\mathbf{y}_t$ towards a constant sensory target $\mathbf{y}^*$. The necessary actions $\mathbf{u}_t$ to approach the target can be computed by minimising the quadratic cost function[1]:

$$J = \|\mathbf{A}_t \mathbf{u}_t + \lambda \operatorname{sat}(\mathbf{y}_t - \mathbf{y}^*)\|^2 \tag{5}$$

for $\lambda > 0$ as feedback gain, and $\operatorname{sat}(\cdot)$ as a vectorial saturation function to ensure that $\mathbf{u}_t$ satisfies the differential motion condition in (1). By computing the extremum $\nabla J(\mathbf{u}_t) = \mathbf{0}$, we obtain the *normal equation*

$$\mathbf{A}_t^\intercal \mathbf{A}_t \mathbf{u}_t = -\lambda \mathbf{A}_t^\intercal \operatorname{sat}(\mathbf{y}_t - \mathbf{y}^*) \tag{6}$$

which exposes the different properties of the three cases relating the dimension of $\mathbf{y}_t$ with $\mathbf{x}_t$.

For $n > m$, the solution to the problem can be obtained from (5) via the right pseudo-inverse of $\mathbf{A}_t$ as follows [17]:

$$\mathbf{u}_t = -\lambda \mathbf{A}_t^\intercal \left(\mathbf{A}_t \mathbf{A}_t^\intercal\right)^{-1} \operatorname{sat}(\mathbf{y} - \mathbf{y}^*) \tag{7}$$

Note that the above motor action will globally minimise (5) (i.e. $\|\mathbf{y}_t - \mathbf{y}^*\| \to 0$), as long as the m feature coordinates in $\mathbf{y}_t$ are linearly independent with respect to $\mathbf{x}_t$. This ensures that the matrix $\mathbf{A}_t \mathbf{A}_t^\intercal$ can be inverted.

For $m > n$, the solution is obtained by solving the normal equation (6) for $\mathbf{u}_t$, which yields:

$$\mathbf{u}_t = -\lambda \left(\mathbf{A}_t^\intercal \mathbf{A}_t\right)^{-1} \mathbf{A}_t^\intercal \operatorname{sat}(\mathbf{y} - \mathbf{y}^*) \tag{8}$$

Substituting (8) into (5) shows that the cost function can only be locally minimised (i.e. $\|\mathbf{y}_t - \mathbf{y}^*\| \to \eta$, for $\eta > 0$). The use of redundant features is useful in practice to cope with intermittent feedback from sensors, such as in the case of camera occlusions or malfunctions.

For the trivial case of $n = m$, the matrix $\mathbf{A}_t$ is square, therefore, the solution is simply obtained via standard matrix inversion $\mathbf{u}_t = -\lambda \mathbf{A}_t^{-1} \operatorname{sat}(\mathbf{y} - \mathbf{y}^*)$.

3 CONTINUOUS MODEL ADAPTATION

3.1 Uncertain Sensorimotor Models

Computing any of the above motor actions requires some knowledge (at least coarse) of the transformation matrix $\mathbf{A}_t$, which in turn, depends on the sensor and the feature models $g(\cdot)$ and $f(\cdot)$. However, if the estimated model is corrupted at some point in time, the robot may no longer properly coordinate action with perception. This situation may happen when the mechanical structure of the robot is altered (e.g. due to bendings or damage of links) or when the configuration of the perceptual system is changed (e.g. due to relocation of external sensors).

The capability to dynamically estimate sensorimotor models is needed to use robots in many growing fields such as domestic robotics, autonomous systems, field robotics, etc., where the sensorimotor conditions are highly uncertain. Several methods have been proposed to compute or approximate these models (see [18] for a comprehensive survey on the topic). In this paper, we coarsely classify these methods into the following two approaches: structure-based estimation and structure-free estimation. In the following sections, we present the model adaptation problem and provide various solutions to it.

3.2 Structure-Based Model Adaptation

These types of algorithms represent calibration-like techniques that aim to estimate the parameters $\boldsymbol{\pi} \in \mathbb{R}^p$ of the uncertain sensorimotor model. Its implementation requires exact knowledge of the analytical structure of the model $\mathbf{y}_t = f(g(\mathbf{x}_t))$, which for ease of presentation, we assume it is linearly parametrisable as[2]:

$$\mathbf{y}_t = f(g(\mathbf{x}_t)) = \mathbf{L}(\mathbf{x}_t)\boldsymbol{\pi} \tag{9}$$

[1]The rational behind the minimisation of (5) is to find a motor command $\mathbf{u}_t$ that projects into the sensory space as a vector pointing towards $\mathbf{y}^*$.

[2]For non-linear model parametrisations, other types of optimisation algorithms must be used, whose details are beyond the scope of this note.

where $\mathbf{L}(\mathbf{x}_t) \in \mathbb{R}^{m\times p}$ represents a *known* regression-like matrix that captures the properties of the analytical model, and whose elements depend on the configuration vector $\mathbf{x}_t$.

To compute the vector of estimated parameters $\widehat{\boldsymbol{\pi}}_t \in \mathbb{R}^p$, structure-based methods require to first collect a set of T observation points $(\mathbf{y}_k; \mathbf{x}_k)$, for $k = 1, \dots, T$ (see e.g. [11]). This data is used for computing the quadratic cost function:

$$U = \frac{\gamma}{2} \sum_{k=1}^{T} \|\mathbf{L}(\mathbf{x}_k)\widehat{\boldsymbol{\pi}}_t - \mathbf{y}_k\|^2 \tag{10}$$

for $\gamma > 0$ as a learning gain. The above function is clearly convex with respect to the error vector $\widehat{\boldsymbol{\pi}}_t - \boldsymbol{\pi}$. Therefore, U can be adaptively minimised with the gradient descent rule:

$$\widehat{\boldsymbol{\pi}}_{t+1} = \widehat{\boldsymbol{\pi}}_t - \nabla U(\widehat{\boldsymbol{\pi}}_t) \tag{11}$$

which in the absence of measurement noise, it globally minimises U (i.e. $\|\widehat{\boldsymbol{\pi}}_t - \boldsymbol{\pi}\| \to \mathbf{0}$, yet, a small estimation error is typically expected in practice). Then, the transformation matrix is then simply computed as:

$$\widehat{\mathbf{A}}_t = \frac{\partial}{\partial \mathbf{x}_t} \{\mathbf{L}(\mathbf{x}_t)\widehat{\boldsymbol{\pi}}_t\} \tag{12}$$

Structure-based approaches have one major disadvantage: its dependency on fixed analytical models. Note that since the model's structure (9) is explicitly used within the adaptation algorithm (11), these methods are not robust to unknown changes in the mechanical and perceptual conditions. Furthermore, in many situations, the analytical model might not be available, or be subject to large uncertainties. This limits the applicability of these types of approaches.

3.3 Structure-Free Model Adaptation

These types of algorithms have the capability to compute the unknown sensorimotor model in the following manner: (i) entirely from scratch (i.e. without requiring any a-priori knowledge of the model's analytical structure), (ii) on-demand (i.e. they can modify its acquired structure at any time instant so as to identify new relations), and (iii) from data observations only (i.e. by using information from controls and measurements only).

Based on its computation, we coarsely classify these algorithms into the following two general categories: *instantaneous* estimation, and *distributed* estimation.

Instantaneous estimation. As the name suggests, these techniques compute a matrix $\mathbf{A}_t$ that is only valid at the current (instantaneous) configuration $\mathbf{x}_t$. The Broyden rule [12] is an example of such technique. It iteratively computes $\mathbf{A}_t$ with the following rule:

$$\widehat{\mathbf{A}}_t = \widehat{\mathbf{A}}_{t-1} + \Gamma \frac{\boldsymbol{\delta}_t - \widehat{\mathbf{A}}_{t-1}\mathbf{u}_t}{\mathbf{u}_t^{\intercal}\mathbf{u}_t} \mathbf{u}_t^{\intercal} \tag{13}$$

for $\boldsymbol{\delta}_t = \mathbf{y}_{t+1} - \mathbf{y}_t$, and $0 < \Gamma \leq 1$ as a tuning gain. With "high" gains $\Gamma \approx 1$, by right-multiplying (13) by $\mathbf{u}_t$ (namely, projecting the motor action into sensory space), we can see that $\mathbf{y}_{t+1} \approx \mathbf{y}_t + \widehat{\mathbf{A}}_t\mathbf{u}_t$ is satisfied. However, using high gains results in a noisy and rapidly changing matrix $\widehat{\mathbf{A}}_t$. For slow robot motions, the Jacobian matrix is expected to change slowly, therefore, using "small" gain values for Γ can help to make the computation less responsive, i.e. $\widehat{\mathbf{A}}_t \approx \widehat{\mathbf{A}}_{t-1}$ as well as to filter out noisy measurements.

Another example of these techniques can be derived from the following instantaneous cost function [19]:

$$V = \frac{\gamma}{2} \left\| \widehat{\mathbf{A}}_t\mathbf{u}_t - \boldsymbol{\delta}_t \right\|^2 \tag{14}$$

which provides a metric of the accuracy of $\widehat{\mathbf{A}}_t$. The terms of this unknown matrix are continuously adapted with the rule:

$$\widehat{a}_{t+1}^{ij} = \widehat{a}_t^{ij} - \nabla V(\widehat{a}_t^{ij}) \tag{15}$$

where the scalar $\widehat{a}_t^{ij}$ denotes ith-row jth-column term in $\widehat{\mathbf{A}}_t$.

With instantaneous estimation techniques, the matrix $\mathbf{A}_t$ must be continuously recalculated with new sensor observations as the robot moves into other configurations. Thus, these types of techniques do not provide a mechanism for locally preserving knowledge of previous estimations.

Distributed estimation. Note that since the feature functional (3) is smooth (i.e. differentiable), its Jacobian matrix is expected to smoothly change along the configuration space $\mathbf{x}_t$. This means that a *local* Jacobian matrix estimated at a particular configuration point is also valid around the neighbourhood surrounding it. This simple yet powerful idea forms the working principle of distributed estimation techniques. With these adaptive algorithms, the estimation problem is shared amongst multiple computing units that specialise in a local transformation.

Consider a network of N computing units spread around the robot's configuration space. The following data structure z^l is associated with each computing unit:

$$z^l = \left\{ \overline{\mathbf{x}}^l \quad \widehat{\mathbf{A}}^l \right\}, \quad \text{for} \quad l = 1, \dots, N \tag{16}$$

where $\widehat{\mathbf{A}}^l$ stands for a local approximation of $\mathbf{A}_t$ estimated at the configuration point $\overline{\mathbf{x}}^l$. There are multiple methods for establishing the distribution of these units around the robot's workspace, e.g. based on self-organising rules, evenly distributed locations, or random point distributions [20, 21]. For ease of presentation, we assume that the location of these N configuration points $\overline{\mathbf{x}}^l$ associated with the units has already been established.

Estimation of local transformation matrices is performed by first collecting a data set of T observation points $(\boldsymbol{\delta}_k; \mathbf{u}_k)$, for $k = 1, \dots, T$. Then, the following local cost function for the lth unit is computed:

$$W^l = \frac{\gamma}{2} \sum_{j \in \mathcal{B}} h^{lj} \left\| \widehat{\mathbf{A}}^l\mathbf{u}_j - \boldsymbol{\delta}_j \right\|^2 \tag{17}$$

for $\mathcal{B}$ as a local ball centred at the lth unit, and h^{lj} as its Gaussian neighbourhood function computed as

$$h^{lj} = \exp\left(-\frac{\|\overline{\mathbf{x}}^l - \overline{\mathbf{x}}^j\|^2}{2\sigma^2} \right) \tag{18}$$

where $\sigma > 0$ determines the ball's radius. In this method, the idea is to make use of neighbouring data (whose contribution decreases with its distance to the centre unit l) for

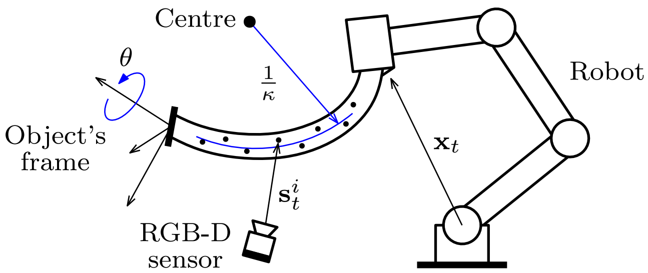

Fig. 1: Shape servoing system composed of a 4-DOF manipulator, an elastic object, and a 3D vision sensor.

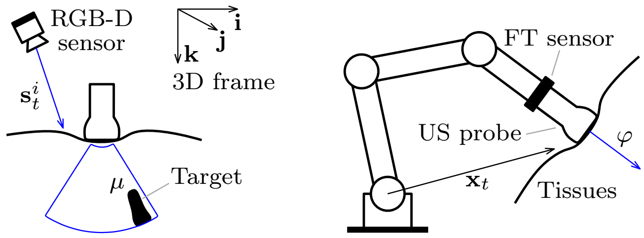

Fig. 2: Multi-modal system composed of a robot manipulator, an ultrasound probe, a force sensor and a 3D camera.

approximating the local transformation matrix. The update rule to adaptively compute the ith row jth column term of $\widehat{\mathbf{A}}_t^l$ is as follows:

$$\widehat{a}_{t+1}^{l,ij} = \widehat{a}_t^{l,ij} - \nabla W(\widehat{a}_t^{l,ij}) \tag{19}$$

Once the N cost functions (17) have been minimised, the network is trained to perform local sensorimotor transformations with each of its units. In order to implement a motor command as the ones derived in Sec. 2.4, the estimated local matrix must be retrieved from the l unit that best matches the current position $\mathbf{x}_t$ by solving the search problem

$$l = \arg\min_j \left\{ \|\overline{\mathbf{x}}^j - \mathbf{x}_t\| \right\} \tag{20}$$

Note that this adaptation approach can be combined with the previous instantaneous estimation technique (or possibly others) by defining a cost function $H^l = V + W^l$ that exploits the current feedback measurements. This allows to also quantify the accuracy of the model based on new sensory data; the cost function can then be minimised with similar gradient descent tools as before.

Compared to the previous estimation approaches, distributed estimation requires considerably more data to approximate the sensorimotor model, however, it can effectively preserve local knowledge within its computing units.

4 CASES OF STUDY

4.1 Visual Shape Servoing of Elastic Objects

To exemplify our methodology, consider the setup depicted in Fig. 1, where a 3D camera captures point clouds of a beam-like elastic object manipulated by a robot. Let us denote the captured 3D points by $\mathbf{s}_t^i$, for $i = 1, \ldots, r$ (note that the number r is generally in the order of hundreds). The task to be performed is to automatically deform the object into a desired shape. We can use the point cloud to approximate a the object's backbone (represented as the blue curve in Fig. 1). With this geometric information, we compute the feature vector defined as follows:

$$\mathbf{y}_t = \begin{bmatrix} \kappa & \theta \end{bmatrix}^\intercal \tag{21}$$

for $\kappa > 0$ as the object's curvature, and θ as the angle of the object's bending with respect to it's frame, see [22] for details. For this task, *model adaptation* can be performed with distributed estimation algorithms. These approaches provide an efficient solution to the highly nonlinear transformation problem of relating robot poses to object deformations (note that deformation models are hard to compute analytically). For that, several computing units must be first defined at key end-effector poses, e.g. ranging from fully stretched to varying bending configurations; local sensory observations can then be collected for approximating the model.

4.2 Multi-Modal Scanning with Ultrasound Probes

Consider the setup in Fig. 2, which depicts a robot performing automatic scanning of tissues with an ultrasound probe [23] (we assume the robot has 6-DOF). This system is instrumented with a force/torque sensor and a 3D camera. Let us denote by φ the (normal) force applied onto tissues, by μ the location of the ultrasound feature of interest, and by $\boldsymbol{\omega}$ the probe's 3D orientation. The task to be performed is conveniently described with respect to the body's 3D frame. It consists in positioning μ at the centre of the ultrasound image, while applying a desired normal force and controlling the probe's pose over the tissues. Note that this relative orientation can be computed from the 3D point clouds $\boldsymbol{\omega} = q(\mathbf{s})$. The task's feature vector is defined as

$$\mathbf{y}_t = \begin{bmatrix} \mu & \varphi & \boldsymbol{\omega} \end{bmatrix}^\intercal \tag{22}$$

The models for the above feature coordinates are simple to analytically derive, namely using Hooke's law for φ, horizontal image displacements for μ, and homogeneous transformations for $\boldsymbol{\omega}$. Therefore, *model adaptation* can be performed using structure-based algorithms as in (9). With these approaches, we can robustify the sensor-guided task by continuously calculating unknown task parameters such as: stiffness of soft tissues, relative location of ultrasound features, and robot-camera-body transformations.

5 CONCLUSIONS

In this expository paper, we addressed the problem of computing adaptive sensorimotor models for robots with uncalibrated sensory feedback and/or uncertain morphology. A general sensor servoing approach was first formulated using an energy minimisation approach. Then, we derived various methods for providing these controllers with continuous model adaptation capabilities. Two cases of study we presented to illustrate the proposed methodology.

The presented sensorimotor controls are formulated based on the assumption that the feedback signals dependent on the robot's configuration only. Although this condition can be fairly used to represent many sensor-guided applications, it may not be the most accurate model for describing tasks where the measurements also depend additional variables (e.g. manipulating fabrics with infinite dimensional configurations) or even time-varying states (e.g. controlling the effect of cosmetic lasers stimulating skin tissues). The development of more general sensor models is still an open research problem.

The presented model adaptation methods allow robots to perform sensor-guided tasks even when its sensorimotor model is not known or might suddenly change. For example, robots can adapt to unknown sensor configurations and/or morphologies. By understanding the principle of how sensors models can be effectively created from scratch and adapted on-the-fly, we hope to build machines with more resilient properties that allow them to perform long-term operations with minimal supervision. These advanced capabilities are needed to advance towards building truly autonomous robots.

References

[1] E. Davison and A. Goldenberg, "Robust control of a general servomechanism problem: The servo compensator," *IFAC Proceedings Volumes*, vol. 8, no. 1, pp. 231 – 239, 1975.

[2] F. Chaumette and S. Hutchinson, "Visual servo control. Part I: Basic approaches," *IEEE Robot. Autom. Mag.*, vol. 13, no. 4, pp. 82–90, 2006.

[3] Q. Li, C. Schurmann, and R. H. R, "A control framework for tactile servoing," in *Proc. Robotics: Science and Systems*, vol. 1, 2013, pp. 1–8.

[4] M. Prats, P. J. Sanz, and A. P. del Pobil, "Reliable non-prehensile door opening through the combination ofvision, tactile and force feedback," *Auton. Robots*, vol. 29, no. 2, pp. 201–218, 2010.

[5] S. E. Navarro, B. Hein, and H. Wörn, "Capacitive tactile proximity sensing: From signal processing to applications in manipulation and safe HRI," *Soft Robot.*, pp. 54–65, 2015.

[6] A. Magassouba, N. Bertin, and F. Chaumette, "Audio-based robot control from interchannel level difference and absolute sound energy," in *Proc. IEEE Int. Conf. Intelligent Robots and Systems*, 2016, pp. 1992–1999.

[7] D. Navarro-Alarcon, Y.-H. Liu, J. G. Romero, and P. Li, "On the visual deformation servoing of compliant objects: Uncalibrated control methods and experiments," *Int. J. Robot. Res.*, vol. 33, no. 11, pp. 1462–1480, 2014.

[8] D. Navarro-Alarcon and Y.-H. Liu, "Fourier-based shape servoing: A new feedback method to actively deform soft objects into desired 2D image shapes," *IEEE Trans. Robot.*, vol. 34, no. 1, pp. 272–1279, 2018.

[9] R. Pfeifer and C. Scheier, "Sensory-motor coordination: The metaphor and beyond," *Robot. and Auton. Syst.*, vol. 20, pp. 157–178, 1997.

[10] G.-Q. Wei, K. Arbter, and G. Hirzinger, "Active self-calibration of robotic eyes and hand-eye relationships with model identification," vol. 14, no. 1, pp. 158–166, 1986.

[11] H. Wang, Y.-H. Liu, and D. Zhou, "Adaptive visual servoing using point and line features with an uncalibrated eye-in-hand camera," *IEEE Trans. Robot.*, vol. 24, no. 4, pp. 843–857, Aug. 2008.

[12] K. Hosoda and M. Asada, "Versatile visual servoing without knowledge of true Jacobian," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, vol. 1, 1994, pp. 186–193.

[13] I. Kohler, "Experiments with goggles," *Scient. American*, vol. 206, no. 5, pp. 62–73, 1962.

[14] D. Whitney, "Resolved motion rate control of manipulators and human prostheses," *IEEE Trans. Man-Mach. Syst.*, vol. 10, no. 2, pp. 47–53, Jun. 1969.

[15] B. Siciliano, "Kinematic control of redundant robot manipulators: A tutorial," *Journal of Intelligent and Robotic Systems*, vol. 3, no. 3, pp. 201–212, 1990.

[16] A. Cherubini, R. Passama, P. Fraisse, and A. Crosnier, "A unified multimodal control framework for humanrobot interaction," *Robot. and Auton. Syst.*, vol. 70, no. Supplement C, pp. 106 – 115, 2015.

[17] Y. Nakamura, *Advanced robotics: redundancy and optimization*. Boston, MA: Addison-Wesley Longman, 1991.

[18] O. Sigaud, C. Salan, and V. Padois, "On-line regression algorithms for learning mechanical models of robots: A survey," *Rob. Auton. Syst.*, vol. 59, no. 12, pp. 1115 – 1129, 2011.

[19] D. Navarro-Alarcon, H. M. Yip, Z. Wang, Y.-H. Liu, W. Lin, and P. Li, "Adaptive image-based positioning of RCM mechanisms using angle and distance features," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2015, pp. 5403–5409.

[20] T. Kohonen, *Self-Organizing Maps*. Springer Berlin Heidelberg, 2001.

[21] S. Haykin and S. Haykin, *Neural Networks and Learning Machines*, ser. Neural networks and learning machines. Prentice Hall, 2009, no. 10.

[22] D. Navarro-Alarcon, O. Zahra, C. Trejo, E. Olguin-Diaz, and V. Parra-Vega, "Computing pressure-deformation maps for braided continuum robots," *Frontiers in Robotics and AI*, vol. 1, no. 1, pp. 1–8, 2019.

[23] M. Victorova, D. Navarro-Alarcon, and Y. Zheng, "3D ultrasound imaging of scoliosis with force-sensitive robotic scanning," in *Int. Conf. Robotic Computing*, 2019, pp. 1–4.