B310 Anchored Robots. An introduction

Javier Finat

Índice

Previous remark.- These notes correspond to a provisory draft of the introduction to the module B_{31} (Anchored Robots) of the matter B_3 (Robotics). It develops a mathematical-based approach to basic topics of Anchored Robots, with a special attention to sequential robots, and one chapter dedicated to parallel robots. Industrial and biomedical applications are developed in the two last chapters.

It is convenient to have a basic knowledge of Basic Algebra (Matrix Calculus and Groups), Differential Analysis (including Optimization) and Differential Geometry. Nevertheless, along the draft one includes references for the main results. From the computational viewpoint, one supposes the reader is familiar with basic notions of Computational Geometry B_{11} and related OOP framework.

As usual, materials are organized in four sections to be given along one month (one per week). They contain a list of exercises for self-verification of understanding of materials. In addition, there is a fifth section is devoted to some complements. Subsections or paragraphs marked with an asterisk ($*$) display a higher difficulty and can be skipped in a first lecture.

0.1. Introduction

The notion of robot is by itself ambiguous because many devices with varying degrees of autonomy are called robots. Following the old definition given by the Robot Institute of America (1979), a *robot* $\mathcal{R}$ is a multi-functional reprogramable manipulator which is designed to move materials, pieces or specialized devices, through variable programmed movements for the realization of different tasks.

In particular, a robot includes mechanical components (acting as a skeleton $Sk(\mathcal{R})$), sensors and actuators (for capturing information and generating responses), and mechatronic devices (for connecting above components and for optimization and control, e.g.). All of them are managed by expert systems with reflex and supervised components for reactive or supervised tasks execution.

The tasks supervision is performed by a central unit and managed by using distributed systems along the architecture. Distributed systems are in charge of coordinating and generating semi-automatic responses (including reflex and voluntary movements). Along the current module B_{31} Anchored Robots), we will be focused mainly towards robots with a fixed basis, with main applications to industrial and biomedical robots. Next issue concerns to the title of this module which is also ambiguous:

Anchored Robots can be understood in different ways, depending on the adopted framework. In industrial environments a commonly accepted meaning concerns to the fixed character of their basis w.r.t. the ground or a wall, e.g.; typical examples of related tasks would correspond to welding, cutting and assembly operations, e.g.. An alternative approach emerging from the early years of the 21st century concerns to heuristic connections between internal representations (linked to symbols as weighted dynamical graphs, e.g.) for physical objects with execution of tasks in the external world.

The second interpretation is nearer to the so-called *Perception-Action Cycle* (PAC in the successive) a transversal topic for almost all the modules developed here; see below. The use of symbolic representations (graphs for the skeleton $Sk(\mathcal{R})$, simplified PL-representations of the scene and tasks, e.g.), and their extension to dynamic issues under uncertainty conditions is a complex domain, where they appear a lot of aspects concerning Knowledge representations, High-level Sensor Planning, Simulations of Functional aspects of Human Brain, and different kinds of Learning procedures, between others.

Thus, due to this higher complexity and to bound the contents of this module, we will be focused towards the first "more industrial" meaning. This choice does not imply some small developments connecting complementary viewpoints (top-down vs bottom-up approaches, "deterministic vs random", different kinds of learning, e.g.) as we will expose at the end of this introduction.

In practice and due to the very high theoretical and practical complexity in Robotics, an advantage of simulation consists of it is not necessary to understand all aspects and reas appearing in Robotics. In particular, the simulation allows to isolate those which are more meaningful for some stages. This simple remark explains the ubiquity of simulation-based techniques for all morphological and functional robotic aspects. The same principles can be applied for learning issues in Robotics.

According to the precedente remarks and furthermore this introductory chapter, the materials of the module B_{31} are organized in nine chapters:

1. *Mechatronic Design* involving geometric aspects, sensors and actuators
2. *Planar robots* acting on the plane or self-adjusting to planar or developable surfaces.
3. *Volumetric robots* acting on the Cartesian space $\mathbb{R}^3$ in industrial environments.
4. *Control strategies* for planar and volumetric robots, including basic robust vs adaptive strategies.
5. *Simulation* involving an introduction to basic strategies to represent rigid motions in Computer Graphics.
6. *Machine Learning for robots* where one makes a short revision and adaptation of the most commonly used ANN for anchored robots
7. *Parallel robots* with Stewart platforms as central paradigm in terms of "configurations" on homogeneous spaces (Grassmann vs Flag Manifolds).
8. *Industrial applications* with a short survey for main tasks developed in industrial environments.
9. *Surgery assistance* with an specific analysis of hyper-redundancy, stiffness and extreme accuracy issues.

Along the next paragraphs one introduces some remarks to understand possible contributions of the approach performed along these notes.

0.1.1. Optimal Design

A central problem for any kind of robots is the study of *Optimal Design* to accomplish tasks in a so efficient way as possible. Biological inspiration shows a lot of efficient examples which have been achieved to improve the performance of tasks after several millions of years of evolution. Thus, even for simplest tasks, the robotic design is linked from the beginning to biologically inspired architectures to imitate main tasks concerning to manipulation and motion analysis (to be performed along each task). To improve their efficiency and adaptability to eventually changing conditions, one requires an optimal design for control of robot devices.

Optimal Design involves to all layers (geometric, kinematic, dynamic) of Robot Mechanics. In particular, manipulation includes motion analysis and control issues, involving Kinematics and Dynamics, respectively. The feedback between all layers is formulated in differential terms (discrete vector and covector fields, e.g.) and integral terms (accumulative evaluation in terms of convolutions, e.g).

Variational Calculus provides a natural feedback between differential and integral aspects. Descent of Kinematic and Dynamic constraints must be taken in account for Geometric Design, and inversely; lifted vector and covector fields from the base space B (a piecewise smooth manifold M to start) provides the key to understand the relevance of Geometric Design on M for Kinematics on the Phase space $P = TM$ and Dynamics on TP. Thus, Optimal Design is implicit along the whole first module B_{31}.

From a mathematical viewpoint, a general solution of Optimal Design is a very difficult task by different reasons: The choice of weights (not easy to identify) in the cost function, the presence of tasks constraints involving maps which must be translated to the support, and constraints at different mechanical levels (geometric, kinematic, dynamic) can be antagonistic between them. An adaptive approach able of bounding uncertainty would must use a topological approach before applying metric criteria. If we reason in a topological way,

- Relative weights $(\lambda_i)_{i\in I}$ for architectural components or functions represented as $\sum_i \lambda : if_i$ can be interpreted in terms of barycentric coordinates or affine coordinates, i.e. $\sum_{i\in I} \lambda_i$ fulfilling $0 \leq i \leq 1$ if they are barycentric coordinates.
- The duality between PL-structures represented by polytopes K and linear functionals $\varphi : K \to \mathbb{R}$ provide a natural feedback between simplified representations for morphological issues (relative to the support) and functional issues (relative to tasks to be performed).
- Finally, the representation of solutions for systems of (geometric, kinematic, dynamic) equations in terms of topological fibrations $E \to B$ on a base space (corresponding to a manifold M, the Phase space P or its tangent space TP) provide the support for lifting and descent properties between superimposed structures.

The foundations for all these issues have been developed in several modules of the matter A_2 (Algebraic and Geometric Topology) involving Homotopy A_{21} (for lifting and descent properties), Homology and Cohomology A_{22} (fitting weights involving components or linear functionals $f : K \to R$) and Topological Fibrations on Cell complexes A_{13} (to incorporate solutions of systems of equations and their behavior by deformations).

As usual, we follow an increasing order of difficulty for the Optimal Design of robots, by starting with planar robots composed of once a kinematic chain operating in a fixed plane (Chapter 2): They can be considered as a planar mechanism of bars which are connected between them through rotational joints, with the snake as a typical example. In some cases, one can incorporate prismatic joints which can modify the length for some bars, generating instaiblities to be corrected. Allowed modificaitons are initially interpreted as deformations given by homotopies with fixed extremes. In this way, the optimal design becomes a variational problem.

Next, one introduces a topological approach for the design of "volumetric" robots, i.e. robots able of operating in ordinary 3D space (Chapter 3) including spherical joints. Some more difficult cases concern to cooperating robots in the plane or the space, which are introduced at the end of chapters 2 and 3. Volumetric architectures display a high complexity, specially when there appear closed loops involving the robot architecture or the working space. In particular, Stewart platforms provide some of the most challenging examples for dynamic modeling and control, as we shall see at the end of this module.

0.1.2. Tasks as dynamic paths

Each *task* of a control point $\mathbf{c}$ for an *anchored robot* is described in terms of a *continuous path* $\gamma : [0, 1] \to X$ starting at $x_0 = \gamma(0)$ and ending at $x_1 = \gamma(1)$ for each control point in the joints or configurations space $\mathcal{C}$ and in the working space $\mathcal{W}$. Each path γ_i performed by a control point $\mathbf{c}_i$ of a robot $\mathcal{R}$ can be understood as a trajectory $\gamma(t)$ supporting additional kinematic (velocities and accelerations) and a dynamic information (forces and momenta) linked to the curvature functions of allowed trajectories.

Topology of paths provides only the initial support for motion planning. Homotopy A_{21} provides the support arising from Algebraic Topology for the analysis of paths and their deformations. An added value of Motion Planning in Robotics w.r.t. to Algebraic Topology framework A_2 consists of superimposing additional dynamics superimposed to trajectories. A typical example is given by elastic curves which are obtained by minimizing $\int_C \kappa_C^2 ds$

Low-level interplay between kinematic and dynamic issues can be formulated from structural equations of trajectories given by Frenet-Serret formulae. Fixation of extremes and boundary conditions for the 22D case were initially studied by Emmy Noether in the first decades of the 20th century. She introduced Lie methods for PDEs linked to variational problems to identify infinitesimal transformations and their corresponding Lie groups.

By these reasons, a theoretical approach to ideal models for paths or flexible plates are initially described in terms of locally symmetric spaces of variational problems. In view of the troubles linked to this analytical approach, we adopt a more heuristic approach. Simulations will be introduced in more advanced chapters of this module [1]

The rigid nature of each component of a robot imposes that motions at joints are described by rigid motions, i.e. elements of the Euclidean group $SE(3)$ given as the semidirect product $SO(3; \mathbb{R}) \ltimes \mathbb{R}^3$ of rotations and translations in the ordinary space $\mathbb{R}^3$. They are represented by (4×4)-matrices where the first 3×3 box representing a rotation $\mathbf{A}$ is augmented by the column $\mathbf{t}$ representing

[1] A more systematic treatment of Simulation will be performed in the module B_{43} (Animation and Simulation) of the matter B_4 (Computer Graphics).

a translation and the column $(0,0,0,1)$; this presentation justifies the use of affine coordinates $(x,y,z,1)$ to represent transformations of control points [2]

An almost trivial advantage of the matrix notation consists of the following: it allows to compute the trajectory performed by the end-effector of each kinematic chain as the product of a finite number of matrices representing rigid motions, one per each component. One must be careful with mechanical constraints, because a complete circular motion is not allowed (to avoid self-intersections of mechatronic architecture). Furthermore, there are forbidden zones for kinematics and dynamics to preserve stability properties. All of them are represented by structural constraints involving the three levels of Mechanics.

Constraints involving to Kinematics and Dynamics "descend" to the whole architecture, by imposing additional constraints to the allowed motions for a robot. The descent is formally represented by using the usual mechanical hierarchy, where the Kinematics is described on the Phase space $P = TM$ (total pace of the tangent bundle τ_MA of a PS-variety M), whereas structural equations for Dynamics are described on the tangent space TP of the Phase space P.

In the classical framework, mechanical constraints are initially represented by a finite set of analytic equations $f_i(\underline{x}) = 0$ and inequalities $g_j(\underline{x}) \leq 0$ and $h_k(\underline{x}) \geq 0$ on M, P or TP, which define an eventually singular semi-analytic variety X with non-empty boundary ∂X. For variable constraints (as it occurs in dynamic environments for evolving interaction), it is more appropriate interpret constraints as "covectors" (corresponding to differential forms in the smooth framework). A unified treatment variable of vectors (trajectories) and covectors (constraints) is performed in terms of Tensor Algebra. Deep Learning provides the standard framework for learning tensors (by using voting procedures, e.g.).

0.1.3. Control issues

A robot $\mathcal{R}$ can have k control points $_i$ for each task; in some cases, it can perform several tasks in a simultaneous way, one per each kinematic chain (bimanual grasping or walking in more advanced multiarm robots, e.g.). Anyway, complex motions are described by means a *multipath* $\Gamma = (\gamma_1, \ldots \gamma_k)$. Trajectories performed by control points arise as a consequence of actions performed at joints by motors to be controlled by different drivers. Traditional approaches to control issues for once a trajectory are developed in terms of Lyapunov functions.

Each component of the robot $\mathcal{R}$ performs a *rigid motion* (translation or rotation) whose composition gives the observed movement for each control point $_i$. Their coupling display a high complexity which must be described in terms of a *distributed architecture*. A simultaneous control of multiple trajectories performed by different smart agents (corresponding to end-effectors of several ki-

[2] The more compact quaternionic representation is preferable because ot avoids rounding errors; it will be introduced in Chapter 3.

nematic chains, e.g.) requires the evaluation of mechanical characteristics (localization, speed, acceleration) for each control point in terms of generalized coordinates $(\underline{q}, \underline{p}, \underline{r})$ provided by sensors, and the fitting to structural equations.

The joint consideration of all these evolving data gives a vector distribution $\mathcal{D}$ generating a k-dimensional subspace L^k to be controlled. If we interpret L^k as a point of the Grassmann manifold $Grass(k, n)$, one can adapt the Lyapunov method to functions defined on $Grass(k, n)$, which can be represented by $k \times n$-matrices of rank k. In particular, a fixed point for the map defined on $Grass(k, n)$ corresponds to a equilibrium configuration configuration, whereas a stable trajectory corresponds to a stable trajectory. The problem consists of giving optimal control criteria in this framework.

Roughly speaking, optimization issues are described in terms of a set G of constraints involving "cost" (energy, elasticity, curvature, e.g.) and "benefit" (adjustment to an ideal trajectory, e.g.) functions. All of them are described in terms of distance functions in appropriate functional spaces (typically Hilbert spaces, e.g.). The inclusion of more advanced propagation phenomena along the mechatronic architecture requires the use of elliptic operators (extending the basic Laplace's operator) involving functions of "bounded variation". Some more technical conditions are formulated in terms of Fredholm operators (finite-dimensional kernel and cokernel) which have ben studied in last chapters of the module A_{42})Fiber Bundles) in regard to Index Theorems.

In the simplest cases involving only to localization and their tracking, one uses hybrid control functionals given by a weighted sum of L^1 (Manhattan) and L^2 (Euclidean) distances. The Manhattan distance allows to remove outliers in cuboidal representations, whereas the second one is used to finer adjustment based on spheres. They can be applied in a simultaneous or a consecutive way. This approach is compatible with Total Variation methods which are commonly used in other processes, going from Image Analysis and Processing B_{21} in Computer Vision to more advanced variational issues in Computational Differential Topology B_{13}.

0.1.4. Motion planning. A first approach

The main goal in regard the execution of a task is its optimal execution avoiding collisions. A simple task is described by once a path γ which is initially described in the Cartesian space; whereas complex tasks $\Gamma = (\gamma_i)_{i \in I}$ (to be performed by bimanual robots, e.g.) are given by a finite set of paths γ_j for $1 \leq j \leq k$, it is necessary describe them in terms of the *mechatronic structure*. It is composed of:

- a *mechanical architecture* (sometimes called an *articulated-body*) composed by a finite number of kinematic chains with a fixed basis;
- an *electronic architecture* including sensors and actuators, control systems and specific hardware for connecting different subsystems;

- a *software architecture* for coordination, adaptation and reengineering tasks to be performed by each kinematic chain.

A good (usually distributed) design of the mechanical architecture is crucial to improve *typical tasks* such as grasping and handling, welding, painting, cleaning, etc. First industrial robots were designed to perform once a task. Their mechanical architecture was composed by eventually extensible arms (to generate translations) and (rotational or spherical) joints.

Simplest robots are non-redundant, i.e. there is once a way of performing a task; they are typical in early industrial environments. From the eighties, there is an increasing interest in (hyper)redundant robots where each task can be performed in several ways as it occurs with most of live beings. In this case, it is necessary find the optimal path in the space of solutions under internal constraints (involving the mechanism) and external constraints (involving the environment).

2mm

0.1.5. Geometric design and Lie groups

The design of each task requires a careful design able of "translating" motions to be performed in terms of the mechanical architecture, according to mechanical constraints. The problem becomes more complex for "redundant robots" whose number of degrees of freedom (dof) is higher than the dimension of the ambient space. Typical examples are given by robots in assisted surgery, where the task can be accomplished in different ways; in this case, one must consider optimization techniques which will be incorporated at last chapters of this module. Anyway, the mechanical design involves to an adequate relative disposition of joints and bars to ease a transmission of Kinematics and Dynamics; as usual, biologically inspired patterns play a fundamental role. In particular,

- kR robots are composed by k *rotational joints*, where each spherical joint represented as an element of $SO(3)$ is decomposed in three rotational joints. Classically, they correspond to the decomposition of constrained spherical rotations in terms of Euler angles.

- Translations require *prismatic joints*, i.e. components with modifiable length in an allowed range which is parameterized by a bounded region of $\mathbb{R}^3$. Bounded variation for functions must be translated to control functions on the corresponding Lie groups.

Hence, the set of allowed rigid motions is a boundary PS-manifold of a group G which is a finite product of copies for the rotations and translations groups corresponding to joints of a robot $\mathcal{R}$. This bounded regions defines the *Configurations Space* $\mathcal{C}$. The composition of small motions at joints (given as a product of matrices) generates effective motions performed by the end-effector of each arm of a robot.

The set of attainable positions is named the *Working Space* $\mathcal{W}$ for the end-effector for each arm or kinematic chain of the robot $\mathcal{R}$. Both Configurations and Working spaces are related through a "mechcnical transfer map" $\mu_{\mathcal{R}} : \mathcal{C} \to \mathcal{W}$ representing how small motions at joints are translated in effective motions at the end-effector of each kinematic chain. In other words, it is represented by a product of matrices.

From a Geometric viewpoint, due to the existence of constraints for the mechatronic architecture of the robot $\mathcal{R}$, both configurations $\mathcal{C} = (C, \mathcal{O}_C)$ and working $\mathcal{W} = (W, \mathcal{O}_W)$ spaces are initially described in terms of a base space X and regular vector functions on $\mathcal{O}_X$. They are locally given by matrix expressions whose entries are trigonometric functions (up to translations) under constraints (to avoid self-intersection).

The non-linear part of the Euclidean Group $SE(n) := SO(n) \ltimes \mathbb{R}^n$ is described for the Cartesian plane $\mathbb{R}^2$ in terms of the group $SO(2)$ (or $U(1)$ for unit complex numbers). Similarly, for the Cartesian space $\mathbb{R}^3$ in terms of the group $SO(3)$ (or its double covering $SU(2)$ for unit quaternions). In the last chapter, we shall give a description in terms of bivectors given by screws, twists and wrenches in the Geometric Algebra framework.

In more formal terms, configurations and working spaces have initially a structure of a "locally symmetric semi-analytic variety". In other words, it is a subvariety of a product of a finite number of copies of Euclidesn groups $SE(2)$ and $SE(3)$, where equations are locally given by a finite number of equalities and inequalities (corresponding to constraints). So, finite set of analytic functions (convergent Taylor development) define a semi-analytic subvariety of a product of Lie groups . Hence, the transfer map $\tau_{\mathcal{R}}$ is a semi-analytic map which can be locally described in terms of projections or more generally "submersions"

In the smooth case, a *submersion at* $x \in N$ is a map $f : N \to P$ between manifolds whose differential $d_x f$ is a surjective map; in other words, the Jacobian matrix has maximum rank equal to $min(n, p)$ at $x \in N$. Submersions appear in a natural way in Robotics, Computer Vision and Expert Systems. In particular, SOM provide a very interesting "example" in regard a drastic reduction of dimensionality in a regular way [Koh97] [3]

The dual notion of submersion is given by *immersion at* $x \in N$; in this case the differential $d_x f$ is injective. A deep result of M.W.Hirsch in the seventies shows that the topological classification of immersions is equivalent to the homotopy classes of maps between their differentials. By duality one has a similar result for submersions. These results can be justified by regularity conditions of the differential (in local terms, the Jacobian has maximal rank). For more general cases, it is necessary to consider non-vanishinc $Ker(d_x f)$ and $Coker(d_x f)$, and maps between them.

From and algebraic viewpoint, automatic learning can be reformulated in terms of maps between Graded Complexes corresponding to successive maps between Modules (one per each layer). Standard techniques of Homological Al-

[3] T-Kohonen: *Self-Organizing Maps* (2nd ed), Springer-Verlag, 1997.

gebra developed in the chapter 4 (Resolutions) of A_{23} (Cell Complexes) allow to extend immersions and submersions to injective and projective resolutions. These constructions extend the theoretical framework for classifying submersions and immersions to more general non-regular situations appearing in meaningful Engineering areas (Computer Vision, Robotics, Artificial Intelligence).

In basic applications, motions at joints (described also as lifted paths $\tilde{\gamma} : [0,1] \to \mathcal{C}$ of paths defined on the working space $\mathcal{W}$) can generate vibrations at joints and instabilities for the mechanisms. Thus, it is necessary to compensate them by means other components and stabilization mechanisms. In particular, furthermore inertial effects linked to motion, an arm with variable length generates a displacement of the center of gravity (c.o.g.) $\mathbf{G}$ of the robot. Hence, the design must include components to generate anticipatory and compensatory effects. A symmetric design eases the management of dynamical effects.

The most stable anchored robots with prismatic and spherical components are given by Stewart platforms (Chapter 7). They are composed by two polygonal platforms whose vertices are connected between them by extensible bars with spherical joints at their extremes. These parallel robots display higher complexity than arm robots. Therefore they are considered at the last chapter of this first module B_{11}. Thus, in most developments we are restricted to serial manipulators given initially by once an arm connected to fixed platform to the ground.

Typical pick-and-place tasks for a robot include grasping and handling rigid objects. Along this module, we consider only *rigid* objects, i.e. distances between control points remain fixed by the action of rotations and translations. Both types of transformations generate the Euclidean group $SE(n) := SO(n) \ltimes \mathbb{R}^n$ acting on the n dimensional Euclidian space $\mathbb{E}^n$.

Forward problems involve to the composition of matrices representing rigid motions. Inverse problems concern to the identification of the most appropriate combination of motions. Our strategy for their estimation is similar to the used in 3D Reconstruction B_{22}. It is based on the decoupling between rotations and translations, and the estimation of generators for the Lie algebra $\mathfrak{so}(3) := T_I SO(3)$ whose generators are much easier to compute by using a SVD strategy. The exponential of the element $X \in \mathfrak{so}(3)$ gives the requested element $A \in SO(3)$.

The above reasoning must be applied to each spherical joint of a kinematic chain (for rotational joints is almost immediate). After solving it for each kinematic chain, one obtains the whole *articulated-body motion* by composing the effect for each kinematic chain as a path $\gamma : I \to SE(n)$ in $SE(3)$ for the center of gravity G where $\gamma(0)$ corresponds to the initial configuration and $\gamma(1)$ is the final configuration.

If the robot has k arms (typical in multi-arms robots, e.g.), their motions are described by a multi-path, i.e. a finite collection $\Gamma = (\gamma_1, \ldots, \gamma_k)$ of k paths. To fix ideas, we shall consider initially ordinary paths instead multi-paths. Typical examples correspond to bimanual robots such those use in robots for assisted surgery or for manipulation of dangerous or toxic materials.

To ease programming tasks, very often one takes PL paths γ (PL: Piecewise Linear). However, this choice is misleading in regard to the apparition of singularities at corners of polygonals. Furthermore, along each segment, one must design activation-inhibition patterns for signals according to "plateau functions" given by pairs of hyperbolic tangent, e.g.. Plateau functions display two inflection which require an adaptive control approach.

The above remarks, illustrate how smooth and algebraic frameworks provide two complementary approaches which are very useful for control issues (including the analysis at singularities), and a global study of families of solutions for problems involving geometric design, kinematics and dynamic issues. All of them pose interesting non-trivial problems specially in regard with complex interactions even for anchored robots.

Furthermore parallel robots (with Stewart platforms as paradigm), the most interesting problems concern tobimanual operations. In fact, a biomechanical model for the upper half of the human body is given by a Stewart platform for the trunk and a bimanual robot. Some of the above arguments were refined from fruitful talks with Dr. Zhicheng Hou along my stay in the GIAT (Spring, 2020) where both robots (Stewart platforms and bimanual robots) were available. My sincerest thanks again for their invitation, their hospitality and giving me the opportunity of discussing with him some previous materials. Anyway, mistakes or misunderstanding arising from these talks are only mine.

0.2. Outline of the module

Along the next paragraphs, one gives some snapshots to understand our choice of materials and a unifying approach to the main methods. In particular, we have followed a hybrid strategy which is based on a feedback between top-down and bottom-up approaches. This strategy is applied to each one of the sections of this chapter labeled as

1. Some motivations and applications.
2. Mechatronic architecture.
3. Some mathematical contributions.
4. ANN based modeling.

Roughly speaking, along the two first sections the bottom-up strategy is more relevant, whereas along the last two sections, a top-down approach is privileged, which requires some adaptation of the current Bible for DNN [Goo16] [4].

In the following paragraphs we include some comments appearing in the introduction of each section.

[4] I.Goodfellow, Y.Bengio and A.Courville: *Deep Learning*, The MIT Press, 2016

0.2.1. Some motivations and applications

Simplest initial anchored robots are planar or follow cartesian patterns (allowing displacements in orthogonal directions). Manipulation along cartesian axis are useful for automated management in logistics, where they can be coupled with mobile platforms, e.g.. Some applications of Cartesian robots are focused towards management operations, in regard to advanced logistics, distribution, and/or pick-and-place operations in large environments for containers in harbors or railway stations, e.g.. They can operate 24h/24h, to perform inspection tasks from elevated platforms, exchange elements in docklands or large repositories for merchandises. Efficient optimization strategies are crucial to improve terrestrial or maritime international trade. Our approach uses Petri nets.

More complex anchored robots for industrial applications are not planar ones, i.e. they operate in a three-dimensional ambient space, requiring at least a spherical. The maximal number of d.o.f. of a space robot is six: it can be describd in terms of rotations and translations corresponding to the Euclidean group $SE(3) = SO(3;\mathbb{R}) \times \mathbb{R}^3$ or in terms of the 6-dimensional space of lines in 3D. In the Geometric Algebra framework, one can use quaternions, including its matrix version (in terms of Pauli matrices, e.g.).

Next, we consider a more advanced case of intended versatile robots which are being introduced under the label of military robots. In fact, usual related operations are presented as civil operations accordingly to the policy of Army which is usually disguised in civil terms as a cooperation or protection of civil population. Typical applications include demining, construction of civil infrastructures in hostile environments, undersea exploration, sampling extraction and/or protection of critical installations. Embedded intelligence for this kind of mechanism poses hard challenges which require additional research.

Materials are organized around four subsections labeled as (1) Industrial Robots, (2) Cartesian Robots, (3) From Military to Civil Robots, and (4) Exterior Space Robots.

0.2.2. Mechatronic architecture

Embedded hardware in Robotics is organized according to a hierarchy involving different layers:

- *Drivers layer* corresponding to Sensors and Actuators, the electronic circuitry and their integration in a commonly shared *platform*
- Information *Processing and Analysis* for path planning, avoid collisions,
- *Mechanical Layers* for natural hierarchies involving Geometry, Kinematics and Dynamics
- *Advanced tasks* related to steering, teleoperation, cooperative behavior

Inputs provided by sensors are stored in data layers which must be read by higher level algorithm layers. After reinterpreting the low-level information arising from sensors, in terms of Processing and Analysis tools, they must re-converted in commands to be performed by the Action module supporting the Action Space $\mathcal{A}$. The specification of an adequate hardware configuration for the *platform layer* connecting the driver layer (first subsection) and algorithm layer (second subsection) is key to improve the performance of the system.

Materials are organized around for subsections labeled as

1. *Drivers Layer* including pettern recognition of signals (by combining statistical and differential approaches)

2. *Real-time Information Processing and Analysis*, including the noise minimization and packaging information in models.

3. *Mechatronic hierarchies* to ease the information fusion (performed in the space-time domain, instead of the frequency domain).

4. *Advanced communications tasks* including modulation, normalization and their re-mapping to be used by other components.

Along the first chapter $B311$ one gives a more detailed presentation.

0.2.3. Some contributions of our approach

The main mathematical contributions developed along these notes are organized around several topological notions which are linked to topological G-fibrations to reduce information, stratifications for hierarchies and the use of different kinds of fields for control drivers. All of them provide a support to represent relations between base spaces, fibers (corresponding to added information) and the total space of fibrations between spaces.

A topological approach provides a support for a natural incorporation of uncertainty linked to the estimation of parameters, tracking of their evolution, and correction of results by means the application of different kinds of mechanical devices. In particular, homotopy methods allow to fix initial and final states of an evolving system, and control the variation in terms of bounded (scalar, vector, covector) fields to adapt the behavior to the required conditions, by avoiding troubles due to a fast variation of mechanical quantities.

Homotopies are commonly used in Engineering from long time ago, in terms of continuous deformations which are sometimes labeled as "continuation methods" which is one of the simplest types of interpolation $(1-\lambda)f+\lambda g$ between two functions for $\lambda \in [0,1]$. They provide a theoretical support for advanced interpolation methods, which are parameterized by families parameterized by the interval $[0,1]$ or, in presence of closed loops by $\mathbb{S}^1$.

The computation of homotopy groups provides a framework for computing the number of connected components (corresponding to the 0-th homotopy group $\pi_0(X, x_0)$) or, more generally, "path classes" depending on the topology of the support given by a base space B (corresponding to the first homotopy group or fundamental group $\pi_1(X, x_0)$). Less frequent is the use of higher homotopy groups, which is associated to maps $f : \mathbb{S}^n \to X$ corresponding to some class of "normalized data" (parameterized in terms of a topological sphere) [5].

The superposition of additional information (including semantic maps, e.g.) is formulated in terms of topological fibrations, which is an extension of covering spaces, a common topic in Homotopy Theory. Homotopy methods are commonly used in Engineering as continuation methods, even if they have not in account the global topology of the ambient space, usually. However, the notion of topological G-fibrations is less known in Engineering. Roughly speaking, it consists of "covering maps" with a group action on the fiber, where the covering map verifies additional conditions relative to "local topological triviality" to propagate information according to deformations given by homotopies on the base space

More precisely, let us consider a continuous map $p : X \to Y$ (typical examples appear linked to structural projections of bundles, e.g.). One says that $f : Z \to Y$ is covered by p if there existe a mapping $g : Z \to Y$ such that $f = p \circ g$ (typical examples are given by sections of a bundle, such those appearing for detectors or descriptors in Expert Systems). We are interested in the behavior of f by deformation. Thus, lets us consider a homotopy $F : Z \times I \to Y$ where $I = [0, 1]$. such that $f(x) = F(x, 0)$ is covered by $g : Z \to X$. In symbolic terms one has a commutative diagram

$$\begin{array}{ccc} & & X \\ & \nearrow & \downarrow \\ Z \times I & \to & Y \end{array}$$

With the above notation, the map $p : X \to Y$ is a fibration if given any space Z and any homotopy $F : Z \times I \to Y$ whose initial map $f(z) = F(z, 0) : Z \to Y$ is covered by a map $g : Z \to X$, the whole homotopy F "down below" in Y is covered "un above" in X by some homotopy $G : Z \times I \to X$, i.e. $p \circ G = F$. In this case, G is called a covering homotopy for F with initial map g.

This construction taken (from [Nov96][6], p.21) is meaningful in several Engineering areas, because it provides a structural framework to lift and descend information between a base space and superimposed structures in very general conditions (including the discrete and probabilistic frameworks). It is compatible with uncertainty (only matter initial and final "states" or behaviors), and can be extended to any kind of fields starting from a deformation Z of the base space Y taking values in X.

However, due to the extreme generality of this construction and the availability of additional information, it is advisable to incorporate "more structure".

---

[5] A detailed presentation can be read in the module A_{21} (Homotopy) of the matter A_2 (Algebraic and Geometric Topology)

[6] S.P.Novikov: *Topology I*, Enc. Math. Sc. Vol.12, Springer-Verlag, 1996.

The additional 'differential vs integral" 'differential vs integral" arises from basic arithmetic operations in the discrete case (differences as estimators of derivatives, weighted sums as estimators for integrals, e.g.). The first subsection is devoted to introduce a coarse approach to stratifications as a paradigm to be used along the whole matter B_3 (Robotics). The next ones are devoted to Symmetries for the control of morphological and functional aspects, and some applications to Collaborative Robots (CoBots) in industrial environments.

0.2.4. ANN based modeling

According to the precedent section, our topological approach to ANN is based on a reformulation of basic elements for recognition issues given by detectors, descriptors and classifiers. All of them are described in terms of "sections" of superimposed structures given by topological fibrations. In particular,

- *detectors* are interpreted as sections of $0D$ or $1D$ fibrations whose fibre is a sample on a curve corresponding to an initially unknown trajectory inside the network, e.g.;
- *descriptors* are interpreted as sections of k-dimensional fibration. In particular, if the fibration is a vector bundle, it can be interpreted as an evolving k-dimensional space (corresponding to multiple trajectories as multivectors or constraints interpreted as covectors). In the discrete case, it can correspond to a "discrete star shaped" polyhedron where edges are managed in terms of graphs e.g.
- *classifiers* are interpreted as equivalence classes of maps appearing in the notion of fibration. In particular, for regular maps (immersions vs submersions) in the smooth case, their equivalence classes correspond to homotopy classes of maps between the corresponding tangent bundles.

The above topological approach provides enough "relaxation models" for any kind of learning models in classical vs modern ANN models. This formulation provides a support which is compatible with uncertainty about parameters, lack of information about "intermediate stages", and independence w.r.t. the number of layers.

Furthermore, it allows the integration of classical SOM for unsupervised learning. A key element to reduce data dimensionality is the use of "skewed projections". In the smooth case, they are given by submersions, i.e. maps $f : M \to N$ whose differential $d_x f : T_x M \to T_{f(x)} N$, represented by the Jacobian matrix at $x\imath M$, is surjective. A linear map $\varphi : V \to W$ is surjective iff $Coker(f) := W/\varphi(V)$.

The notion of local immersion at $x \in M$ is the dual notion of immersion, i.e. it is characterized by $Ker(d_x f) = 0$. Immersions provide a support for matching data. The topological classification of $f : M \to N$ is equivalent to the homotopic classification of $d_x f : T_x M \to T_{f(x)} N$. The proof (given initially by

Hirsch around 1970), uses only maximal rank for the Jacobian matrix. Thus, the same result is true for submersions. This simple reark provides a topological framework to justify stability arguments in unsupervised learning procedures developed in the SOM [Koh97] [7]. Their extension to DNN is almost immediate.

Mathematical models for self-replication and self-organization are commonly used in Mathematics. The first subsection is devoted to summarize some meaningful features in terms of different kinds of symmetries. The oldest precedents for self-replication involve to regular tessellations, already known by Greeks around 2300 years for the cartesian plane $\mathbb{R}^2$ and space $\mathbb{R}^3$ (five regular polyhedra of the Euclidean space $\mathbb{E}^3$).

From a dynamical viewpoint, regular geometric models can be adapted to any surface (in the Conformal Geometry framework) and regular behaviors to regular propagation phenomena (with Bousinesq equation as prototype for planar hexagonal patterns, e.g.); last ones are extended to diffusion-reaction models where the waves propagation includes some interaction with the environment. Geometric replication models are obtained by means the action of a discrete group G on distributions of points. These models are too regular to be realistic.

A natural issue is to ask about the capability of an artificial system to generate self-replicating models, compatible with small irregularities, able of self-adaptation to objects with variable geometry or processes with variable behavior. Crystal growth and heat propagation patterns on regular surfaces provide examples illustrating the simplest cases.

Replication of cellular structures as living organisms provide much more complex and diverse examples which are a source of inspiration for artificial models. In particular, initial cellular automata of Von Neumann provide the first examples for self-replication. Equilibrium vs periodic behaviors in deterministic systems must be replaced by stable vs oscillating structures around some kind of "attractor".

0.3. Basic references

Only classical textbooks and the encyclopedia [Sci08] are included

[Abr84]R.Abraham and J.E. Marsden: *Foundations of Mechanics (2nd ed)*, Addison-Wesley, Benjamin Cummings, 1984.

[Arn78] V.I.Arnol'd: *Geometrical Methods of Classical Mechancis*, GTM 60, Springer-Verlag, 1978.

[Blo03] A.M. Bloch: "Non holonomic Mechanics and Control", Springer-Verlag, 2003.

[Goo16] I.Goodfellow, Y.Bengio and A.Courville: *Deep Learning*, The MIT Press, 2016

[Koh97] T.Kohonen: *Self-Organzing Maps (2nd ed)*,Springer-Verrlag, 1997.

[Lat91] J.C.Latombe; "Robot Motion Planning", Kluwer, 1991.

[7] T.Kohonen: *Self-Organizing Maps* (2nd ed), Springer-Verlag, 1997.

[Mur94] R.M. Murray, Z.Li and S.S.Sastry: *A Mathematical Introduction to Robotic Manipulation*, CRC Press, 1994

[Sch04] J.de Schutter, and J.Baeten: "Integrated Visual Servoing and Force Control. The Task Frame Approach", Springer-Verlag, 2004.

[Sci00] L.Sciavicco and B.Siciliano: *Modelling and Control of Robot Manipulators* (2nd ed), Springer-Verlag, 2000.

[Sel96] J.M.Selig: *Geometrical methods in Robotics*, Springer-Verlag, 1996.

[Sic08] B.Siciliano and O.Khatib (eds): *Handbook of Robotics*, Springer-Verlag, 2008.

0.3.1. Software resources

Nevertheless its generality and scarce references, it is convenient to read references included in Wikipedia and explore the corresponding links (any suggestion is welcome) [8]

Final remarks: Readers which are interested in a more complete presentation of this chapter or some chapter of this module B_{31} (Anchored robots), must write a message to franciscojavier.finat@uva.es or to javier.finat@gmail.com

[8] https://en.wikipedia.org/wiki/Open-source robotics