UNIVERSITÀ DI PISA

Dipartimento di Ingegneria dell'Informazione

Corso di Laurea Magistrale in Ingegneria Biomedica

Master Degree thesis

Modeling and experimental validation of a parallel microrobot for biomanipulation

Supervisors:
Arianna Menciassi
Sinan D. Haliyo

Candidate:
Arianna Saracino

Academic Year: 2014-2015
17 Luglio 2015

Contents

List of Figures

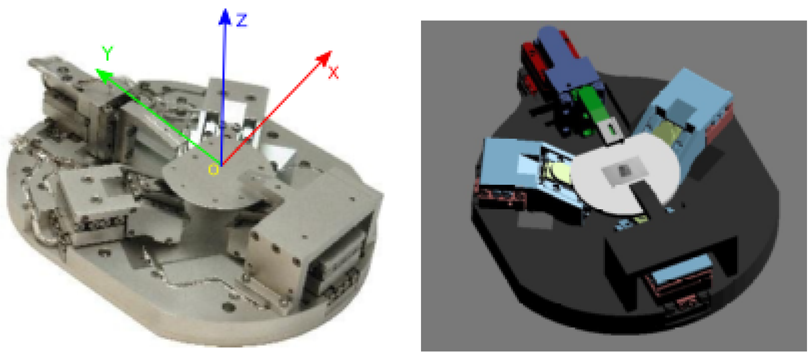
[...] Come, my friends,
'T is not too late to seek a newer world.
Push off, and sitting well in order smite
The sounding furrows; for my purpose holds
To sail beyond the sunset, and the baths
Of all the western stars, until I die.
It may be that the gulfs will wash us down:
It may be we shall touch the Happy Isles,
And see the great Achilles, whom we knew.
Tho' much is taken, much abides; and tho'
We are not now that strength which in old days
Moved earth and heaven, that which we are, we are;
One equal temper of heroic hearts,
Made weak by time and fate, but strong in will
To strive, to seek, to find, and not to yield.
Ulysses, Lord A. Tennyson

Introduction

In the very last decades, the field of **microrobotics** has been object of increasing interest and of a consequent massive technological breakthrough. In fact, investigating, perceiving and fully understanding systems on micrometric and sub-micrometric scales have become critical issues for researchers, involving different areas of application, such as industrial micro-assembly (Bolopion et al., 2013) and micro-manipulation (Haliyo et al., 2005), biology (Ladjal et al., 2011) and techniques for medical science (Mattos et al., 2011).
Reliable and nondestructive manipulation of micro-systems (and of bio-systems more specifically) is required in order to evaluate their mechanical and structural properties (Abrahamians et al., 2013): therefore, robotics has played a major role in developing systems capable of dealing with this characterization.
Disposing of **clear sensors' data** in micromanipulation or microassembly tasks is not easy, as on-board sensors risk to compromise microactuators' performances making systems bulky. In this sense, vision and image processing represent nowadays a valid solution, but a proper instrumentation has to be chosen to observe micro-objects: the view may be obstructed by tools that are orders of magnitude larger than the parts being handled. **Scanning Electron Microscope** (SEM), for example, offers a resolving accuracy of approximately $1nm$, allowing to quantify fine details and parameters of a specimen and overcoming the light-diffraction-limited low resolution of optical microscopy ($200nm$). Moreover, thanks to its high depth of field, SEM technology is able to provide resulting images with a three-dimensional appeareance. SEM has been employed in electrical and mechanical characterization of micro and nano-objects due to the increased room for manipulation it provides, as long as a depth of field and a field of view considerably better than optical microscopies.
Researchers would like to use tools, such as **Virtual Reality** (VR), which enhance micro-manipulation performances, simulating the sample's positioning and possibly helping to create safe and efficient motion's trajectories of the robot itself. Operators who have to deal with the micro-world can also take advantage of VR to see, interact and extract information from it.
In the specific case of SEM environment, VR simulation helps to explore positioning, manipulation strategies and trajectory planning of tasks without risking neither the microscope's internal set-up nor the delicate microrobot architecture (Sauvet et al., 2012):

once the manipulation task is optimized through several simulation attemps, it can be executed easier and faster on the real robot through a proper experimental set-up. Moreover, inside a SEM chamber, it's impossible to have a clear vision of the robot, as SEM's own instrumentation only provides a sample's magnification, while robot's dimensions are considerably bigger than the sample's ones and can't be visualized through the microscope. Another fundamental role of VR is to provide an exact and definite representation of robot's architecture, which could be useful to visualize, for example, its kinematic behaviour in a user-friendly environment.
In a first instance, one can imagine two possible paths to deal with the micro-world, no matter which field of application or specific task is involved: designing a completely new robotic system or selecting a commercial one.
In the first case, various issues have to be taken into account: micro and macro scales are characterized by different hierarchies of forces, proper actuation/sensing tools have to be chosen and a clear understanding of microfabrication techniques is necessary for everyone embarking in developing a new project in the micro areas.
Buying an already existing robotic system, however, requires a full understanding of its behaviour and a sort of **reverse engineering** approach could be useful, in order to enhance its performance by correcting possible bias-errors. In certain cases, manufacturers do not properly document hardware and control solutions of their devices, whose performances often risk to be below the announced results.

This thesis approaches a particular case study of a commercial robot (shown in fig. 1). It is a vacuum-compatible micropositioner characterized by parallel kinematics and able to work inside a SEM, by positioning and orienting a specimen on its moving platform. The micromanipulator is employed for various purposes, ranging from Micro-Electro-Mechanical Systems (MEMS) mechanical characterization to visual servoing applications (Cui et al., 2014). The system represents, in this case, a **black box** from which the operator has to extract information. Understanding its internal laws of motion, in the first instance, means to obtain a **Geometrical Model** (GM) of the positioning system, with the aim of enhancing real-time control scheme's performances, positioning quality and accuracy. Therefore, it is possible to make more reliable the characterization outcomes on micro-biosystems is possible.
To achieve an accurate GM of such a complex micromanipulator, a detailed analysis of its main features is required, along with a full understanding of its motion patterns and kinematics. To do so, a 3D model of the robot, implemented on a **VR sofware**, completely free and open-source, named Blender (*Blender Foundation*), will be employed in order to simplify the modelisation. In fact, Blender creates a powerful and intuitive communication between the 3D virtual environment available on its interface and a Python code, allowing a simple interaction between the user and the microworld in terms of recovering parameters of interest, applying desired motions and forces/torques,



(a) (b)

Figure 1: In (a) the commercial micropositioner, with the moving stage's reference frame shown, in (b) its virtual 3D model

possibly remotely performing system's control in closed-loop.
A Blender-aided modelisation approach will be compared to a more traditional one, with the final purpose of extracting the micropositioner's **Inverse Geometrical Model** (IGM): once position and orientation are imposed on the positioner's End-Effector (EE), the final aim is to extract **actuators displacements' values** to reach that specific final pose. Availability of such a tool will open the road for further studies concerning the system's dynamics in closed-loop and for addressing typical microscale systems's issues. Moreover, the role of simulation in VR will be asserted, as a valid help to improve task's quality and to aid modelisation purposes.

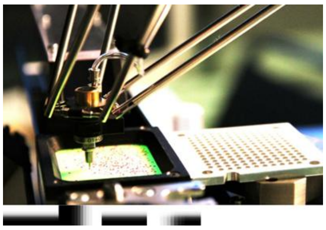
Chapter 1

STATE OF THE ART

1.1 Microrobotics overview

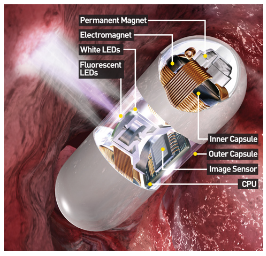
Miniaturization has played a fundamental role in technological progress since the introduction of Integrated Circuits (IC) in the 1950's, calling for both an exponential decrease in the size of electronic components and strong efforts to reduce overall system's dimensions: consequently, exploring the micro-world has been considered a very appealing request in various domains (Liu et al., 2010).
Microrobotics' breakthrough arises from the need for exploring, sensing, manipulating and controlling the world on a smaller and smaller scale. Reducing tools' dimensions allows the investigation of small and clustered environments, inaccessible to traditional macro techniques and equipments. Moreover,the trend towards miniaturization of mass-produced products such as disk drives, wireless communication devices, displays, and sensors is motivating fundamental innovation in design and production.
Microtechnology has been historically defined as a "top-down" discipline, which aims to scale down to micron size traditional mechanical, robotic and control systems (Bohringer et al., 2007).
Microrobotics field, in particular, involves manipulation of all objects whose dimensions range from the millimiter down to the micron scale, as well as the design and fabrication of autonomous robotic agents that fall within this size range. As a general definition, a microrobot performs tasks in the microworld: in other words, tasks in an environment which covers three whole orders of magnitude (1 μ m $\div$ 1 mm). Microsystems can be roughly categorized in the following two classes (Régnier et al., 2010):

- **micromanipulators**, thanks to their submicrometric resolution, are able to perform manipulation of micro-objects, even if robot's dimensions themselves are not necessarily on the same scale. Micromanipulators' end-effectors, on the other hand, have strict size requirements, as they directly interact with the micro-world;



(a) Delta robot (Adept) in microassembly industrial use

(b) Micro-injection system (XenoWorks^{TM}, Sutter Instrument)



(c) Sayaka endoscopic camera pill

Figure 1.1: Main fields of applications for microrobots: industry, biology and endoscopic/surgical fields

- **microrobots** perform direct interactions with micro-objects: being completely immersed in the micro-world, the whole robot has to be micron-sized. They actually take advantage of microfabrication techologies coming from microelectronics: these latter systems are known as MEMS (Micro-Electro-Mechanical Systems.)

In the very last years, increasing interests and consequent enormous advances in microrobotics have lead to the possibility of position and orientation subtle control of micro-objects, allowing the assembly of micro-system. Continuous improvements in micro-fabrication techniques, in robotic research, in the development of autonomous and teleoperated machines, are revealing incredible new perspectives and enabling new applications (see fig. 1.1).
Industry, especially when manufacturing and measuring techniques are concerned, has always dealt with the need of assembling tiny parts with submicrometric accuracy or with the constant challenge between human and automatic handling, with the respective advantages and drawbacks. A typical task, for example, is wafer's inspection, in which several points of it have to be checked with temperature or voltage probes. Generally, activities such as assembly (see fig. 1.1(a)), characterization, inspection and maintenance take advantage of the increasing number of technologies relied with microrobotic systems, as they are characterized by strong accuracy specifications (notably in the sub-micrometer scale).
Also **biology** (fig. 1.1(b))is concerned: considering that applications such as gene research, cell injection or cell sorting require precise, accurate and gentle manipulation of cells, the great contribution of microsystems and microrobotics has already been proved, as will be mentioned in the following sections.
Medicine is one of the disciplines most which exploits microtechnology's improvements: in fact, the role of microrobotics is already spreading in a wide range of medical applications, ranging from catheters to screening/diagnostic/therapeutic wireless capsules

for the gastrointestinal tract, the so called "camera pills" (shown in fig. 1.1(c)), to implantable drug delivery systems and telemicrosurgery. Moreover, Minimally Invasive Surgery (MIS) calls for tiny and flexible endoscopes able to reach target areas in human body through small incisions or natural orifices in order to perform in situ measurements and manipulations. An endoscope, to fulfill all these specifications, should include proper sensors/actuators, microprocessors, a light source and possibily an imaging processing unit (Fatikow et al., 1998): consequently the need for miniaturization becomes crucial for medical-field applications.

Many challanges have to be addressed in order to design and control microrobots' hardware and software, and, more generally, to enhance human interaction with the microscale, starting from **scaling issues**. In the first instance, predominant physics at microscale has to be explored in all its specificities: this is a crucial point, as the most influent forces on micro-system's behaviour are strongly different from those which would be prevalent in an analagous system realized at the macro scale. For this reason, classical modeling of micro-tasks can not be performed without a specific analysis on the involved force. In fact, while at the macro-scale "volumetric" forces such as gravity and inertia are the main ones involved, at the micro scale one will deal with surface, intermolecular and capillary forces, van der Waals and electrostatic interactions. Adhesion forces become predominant when part's size is < 1mm, corresponding to a mass $< 10^{-6}$ kg and they can be due to electrostatic force, van der Waals interaction or surface tension (Bohringer et al., 2007).

Microsystems' performances result highly affected by **environmental conditions**, showing a non linear behaviour due, for example, to slight humidity or temperature variations, to the presence of dust in the working atmosphere or to a chemical composition change in a liquid. The need to work in a controlled environment, thus also preventing unexpected vibrations, becomes essential in order to ensure the micromanipulation's task reliability. It becomes clear that microscopic parts cannot be fabricated, manipulated or assembled in a traditional way: a certain number of assembly steps on the single microcomponent will be required. In this perspective, futher characteristics to be ensured at the micro-scale are: repeatability (less than one micrometer), dexterity (especially for microassembly) and safety (in the case of deformable specimens or in surgery).

The limited availability on board of **sensors and controllers**, because of the small scale of the components, has to be taken into account too. Sensors, in fact, cannot be easily placed on tiny precision instruments without making them bulky or compromising their functionality.

In both micromanipulation and microassembly, however, researchers can rely nowadays on outstanding technologies playing a crucial role in aiding the accomplishment of micro-scale tasks, which can be fully automated or carried out by a human operator. While manual operations call for very skilled and experienced users, automated micro-tasks need an accurate and clear task pre-planning: a model-based approach can be useful in this sense

to define optimal trajectories of manipulation/assembly tasks at the micro-scale, which results also in a reduction of real-time control efforts (Arcese et al., 2012).
In this framework, if one assumes task automation to be more desirable in order to avoid relying only on operator's skills, **Virtual Reality (VR)** provides a unique assistance, as well as intuitiveness, flexibility and an enhancement of user's immersion sensation into the microworld. VR envisages the creation of a 3D reconstruction of the real manipulation/assembly scene, consisting in the robot supposed to perform the micro-scale taks and of micro-objects to be manipulated. Moreover, some advanced application include haptics too, providing a force feedback to the end-user through the interface, in order to increase immersive feeling and to let him/her fully understand the non-intuitive micro-scale physics.

1.2 Sensing and Actuation at the Microscale

Perceiving the world at the micro-scale is quite hard task to carry out. For both sensing and actuation, researches have to address many challenges and different solutions have been tried in order to achieve the best trade-off between performances, reliability, robustness, load capacity, precision, etc.

1.2.1 Sensing

Conventional sensing macro-scale methods are not suitable for microrobots: generally, vision is unanimously considered a suitable sensing tool, because traditional sensors, such as encoders, can not be embedded in micro-systems without affecting their performances. Important factor to take into account in selecting a vision sensing tool at the microscale are its *resolution, depth of field, contrast and brightness* (Nelson et al., 1998), whose influence have to be weighted over the particular application: so also specimen's characteristics are important parameters to consider during the choice.
A general overview about the most employed vision systems to perform micro-scale sensing is provided below:

Optical Microscopy (OM) It allows a magnification up to 1000 times, but several issues associated to it have to be kept in mind while performing sensing at the microscale, such as small depth of field, limited field of view and strong sensitivity to illumination. Moreover, the provided resolution does not exceed 200 nm, preventing it for nano-scale evaluation. OM solution remains the most suitable for in vivo applications (Ghanbari et al., 2009, Ouyang et al., 2007, Ladjal et al, 2011) where air or liquid are required to guarantee specimen's survival.

Electron Microscopy Scanning Electron Microscope has been proposed in particular as a valid alternative to the classic OM in order to overcome issues mentioned above. In fact, SEM offers an infinite depth of field which gives to images a three-dimensional appearence and a resolving accuracy of approximately 1nm: it has been employed in a first istance for the imaging of micrometer-sized structures, in particular to analyse their surface morphology (Eda et al., 2003), while nowadays its characteristics make it a valid equiment to carry out visual tasks, achieving real-time micromanipulation tasks' automation or the investigation of mechanical and electrical properties of nanomaterials (Abrahamians et al.,2013). However, as far as the analysis of biological samples is concerned, additional drying treatments are required when SEM has to be employed, as its chambers are set under High Vacuum (HV), to reduce electron beam's disturbance for observation. Most SEM specimen are imaged at ambient temperature and they must be chemically fixed, dehydrated and eventually coated with a conductive material (e.g. gold) to prevent charge build-up from the electron beam (Kaminskyj et al., 2008). With the E-SEM (Environmental SEM), also known as Variable Pressure SEM (VP-SEM), one can achieve a direct observation of bio-samples while keeping the same nanometric resolution.

Atomic Force Microscope (AFM) Considered a spin-off of the Scanning Tunneling Microscope (Nelson et al., 1998), AFM permits a physical raster scanning to analyse surface characteristics of the sample, taking advantage of the interatomic forces between a cantilever's tip and the substrate. Cantilever's deflections are detected through an optical lever (a laser reflects from the cantilever surface to a photodiode) and the AFM probe can also be used to carry out manipulation tasks (Régnier et al., 2010). AFM provides the same resolution of the SEM and is able to image biological, non-conductive samples in their physiological conditions. Measures are effective under various environments, so using the vacuum is not a compulsory condition such as in SEM. However, in this case, issues related to the humidity-due water film arise, as critical capillary forces originate.

Hybrid solutions and others These solutions have been proposed in works such as (Nakajima et al., 2011), in which authors propose a novel nano-injection system based on E-SEM nanorobotic manipulation. In fact, they combine the OM and E-SEM to realize biological specimen analysis and nano-scale manipulation at the same time. A hybrid solution takes advantage of both the superficial high nanometric resolution of bio-samples typical of the E-SEM and of the inner structure's visualization of transparent samples thanks to OM.

1.2.2 Actuation

One major requirement in the microscale is **high positioning resolution**, along with very **high repeatability and precision**. In this sense, there's a strong need to use

Types of Microscope	Imaging Resolution	Imaging Features	Imaging Environment	Fluorescent Imaging
OM	~200 nm	Interior Imaging	Water/Air	Good
AFM	~1 nm	Surficial Imaging	Water/Air/ High Vacuum	Difficult
SEM	~1 nm	Surficial Imaging	High Vacuum	Difficult
E-SEM	~3.5 nm	Surficial Imaging	Water-Containing	Difficult
TEM	~0.1 nm	Interior Imaging	High Vacuum	Difficult
Hybrid Microscope	~3.5 nm	Interior and Surficial Imaging	Water - Containing	Good

Figure 1.2: Comparison between different micro-sensing technologies

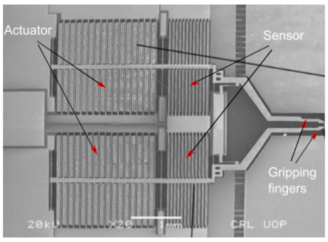
specific actuators, and, in particular, the most adequate ones are those based on active materials, such as piezoelectric ceramics. Their deformable characteristics make them particularly suited to drive microrobots, as they do not undergo mechanical friction as much as traditional actuators. However, non-linear behaviour and hysteresis have to be taken into account by a proper actuators' modelisation (Rgnier et al., 2010).
Actuating sub-millimiter scaled robots is still a significative challange for researches, who have investigated various actuation principles and explored different physical phenomena and energy conversion techniques, in order to achieve the most suitable actuation solution (Dario et al., 1999). Trade-offs among large strokes, high forces, power consumption, system's reliability and robustness, etc. must be examined while thinking about an actuation strategy (Liu et al., 2010).
Main technologies employed to generate motion at the microscale are **electrostatic, electromagnetic and piezoelectric** transduction (see fig. 1.3). Scaling analysis have proved that these actuators have comparable force scaling of $F \propto L^2$ (if permanent magnets are used in magnetic actuation), so it is not possible to select an actuating principle rather then another relying only upon scaling considerations. However, piezoelectric actuators exhibit the significative advantage of possessing a two orders of magnitude larger torque amplitude if compared to electrostatic and magnetic motors. Anyway, as different specifications are needed for each particular application, the best engineering solution should be to select the appropriate actuator case by case.

Electrostatic Since micromachining technology was introduced in 1980s, many types of mechanical actuators driven by electrostatic force have been studied. The motive force employed in electrostatic actuators arises from an attractive or repulsive interaction between oppositely or similarly charged objects (usually plates, such as in a capacitor): in fact, an electric field is established by any electrical charge and it applies a force on any charged particle. Electrostatic force and distance are inversly scaled: great forces are exerced on very short distances, while if the electric field has to work over larger

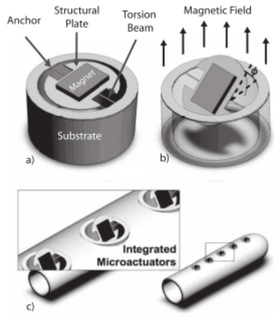
distances, a higher voltage supply is required to mantain a given force. High voltages are easy to get on a large device, but not on a very compact integrated system. On the other hand, power consumption is very low, as well as associated currents, thus assuring a high efficiency.
In any case, they're characterized by low power output and torque if compared to other two strategies at the 0.1-1 mm scale, but they provide high speed operation and little interference with temperature conditions, as electrostatic field arises and disappears very quickly.
In order to optimize performances, several design solutions have been tried (for example cantilevers, comb-drives, induction, side-drive and top-drive, Krijnen et al., 2010).
Electrostatics is the most widely used force in the design of MEMS: electrostatic micro-actuators have been investigated far more than any other principle and consequently they exhibit a rich variety of applications, such as microtweezers (Chen et al., 1989), microgrippers (Boudaoud et al., 2014, Bazaz et al., 2011, se fig. 1.3(a)) and precise positioners of neural microelectrodes (Muthuswamy et al., 2005).

Electromagnetic Electromagnetism arises from electric current flowing into a conducting material. High actuation force and stroke are obtained, along with a contactless remote actuation, characterised by low-voltage supply and generally high energy density. These advantages open the road to magnetic actuation's employment in various fields of application, such as implantable MEMS and catheters (Lee et al., 2009) or optical microsystems (Su et al., 2005, see fig. 1.3(b)).
Despite their massive employment in traditional size systems, they still suffer from strong difficulties in the miniature scale: deposing magnetic materials with MEMS fabrication techniques is complex, so a severe material challenge arises. Moreover, a perpendicularity between the current conductor and the moving element is required in order to generate Lorentz force, which is incompatible with planar fabrication techniques. A final aspect to take into account is that the current drive leads to a high power dissipation.

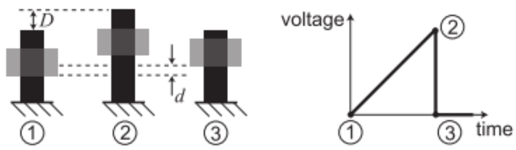
Piezoelectric Displacement in piezoelectric crystalline materials arises from a strain induced by the application of an electric field. Piezoelectric materials present various good features: they operate with high force (10 μN to mN) and high resonance frequency, so fast actuation is achieved with great repeatability. High resolution is also assured, from which derives high precision positioning; moreover, piezoeletrics can also behave as sensors, converting strains into voltages (Liu et al., 2010). On the other hand, some drawbacks are involved too: piezo-actuators provide small strains that must be amplified in order to obtain useful displacement and, in addition, quite high voltages are required. A serious issue is the difficulty of fabricating actuators containing high-performance piezoelectric materials.
A widespread adopted solution in various applications of *long-range* and *ultraprecision*



(a) Microgripper with embedded two electrostatic comb-drive actuators (Bazaz et al., 2011)



(b) Schematic analysis of torsional magnetic microactuators in (Su et al., 2005). In (a) unactuated released state, in (b) actuated state, in (c)the proposed catheter actuated by the previous actuator



(c) Stick-slip working principle

Figure 1.3: Actuation strategies in microrobotics

positioning, in order to overcome traditional piezo's stroke limitation, is the **stick-slip** driving principle: a very simple structure is designed, comprising a sliding mass which moves relative to the piezo element. This configuration confers a very hight stability and stiffness to the whole actuator.
The working principle envisages two modes of operation: the fine one(or step, from phase 1 to 2 shown in fig. 1.3(c)), to reach the highest resolution positioning, and the coarse one (or scan, from phase 2 to 3 in figure 1.3(c)), to reach relatively large strokes. In the first one, a slowly increasing voltage is applied to the piezo element, which stretches continously along with the sliding mass, thanks to the relative friction between them (Nambi et al., 2011). In the second mode, once the system has made the first step (whose length is D), voltage is quickly reversed, causing the piezo to suddenly shrink, while the moving mass slides on it because of the inertia force becoming stronger than the relative friction. This process results in a forward net displacement (d in figure 1.3(c)) of the moving mass respect to its original position: various steps can be accumulated if repeating this procedure, in order to achieve a theoretically unlimited displacement, which is actually only limited by the dimensions of the moving mass itself (Peng et al., 2010). So stick-slip driving principle enables a **high resolution positioning** over a large workspace, along with a theoretically unlimited displacement with a **minute step size** (50 nm), embedding at the same time high stability and stiffness. Moreover, they're often dedicated to automated tasks in vacuum environment, such as SEM (Abrahamians et al., 2013). However, stick-slip actuatos suffer from strong non-linearity (hysteresis, drift and creep) which considerably complicate their modeling; several issues still remain to be addressed.

1.3 The role of Virtual Reality (VR) in micro-scale tasks

A typical micro-scale workstation is composed by computer-controlled devices and components able to fulfill one of the following possible tasks:

- grasping of a target micro-object;
- manipulation/assembly of micro-objects (e.g. pick-and-place tasks);
- micro-objects recognition;
- planning and controlling actuators and end-effectors to accomplish the task.

Working space during one of these micro-scale tasks is generally quite confined: conceiving new micromanipulation strategies and techniques results necessary to cope successfully with *sensing, manipulation, simulation and task planning.*
In this framework, **Virtual Reality** (VR) is asserting itself as an essential and synergical tool in order to overcome, or at least minimize, typical issues related to micro-task

execution, increasing human operators' performances and raising efficiency, repeatability and accuracy of the task itself. VR allows the operator to propose and visualize manipulation/assembly strategies before physically performing them. Moreover, it can be efficiently integrated into a microassembly workcell.
In some cases, **visual servoing** is employed too, ensuring planning and verification of proposed tasks: VR-visual servoing combination increases system's speed and robustness (Cui et al., 2014 - Ferreira et al., 2004 - Cassier et al., 2002).
Creating a VR-based framework helps to address the following typical micro-scale issues.

a) Lack of sensing Sensing at the micro-scale can not actually be just scaled from the macro-scale, as, in most cases, sensors can not be directly embedded in microsystems without affecting their performances or making them bulky.
Image processing has been proposed as a possible solution to compensate for this lack of sensory information, but it is still a slow and expensive technique, very susceptible to reflection and noise.
Visual feedback from optical microscope is limited and does not provide accurate positioning of objects and tools. To improve the quality of the task, it is important that operator disposes of a clear and full field of view of the entire scene and also of various points of view of the manipulated objects: however, with optical or scanning electron microscope, operator's view can actually be obstructed by micro-tools, whose dimensions often are orders of magnitude larger than objects. Moreover, the small microscope's field of view limits the operator, as the small working area can only be seen all at the same time.
VR provides a reconstruction of the 3D scene (an example is provided in fig. 1.4)in which operator can freely move and explore it, avoiding risky situations, such as damaging manipulated objects, suffering from an unstable handling or from misalignments during manipulation. Users, for example, can rely on a global 3D view of the whole system and mistakes related to kinematic singularities, exceeding joint limits or obstacles, can be early detected, before the real task execution.
In a first istance, microassembly workcell can be programmed off-line and rely upon CAD-CAM available data concerning microcomponents, in order to realize a task planning. Later, employing computer-vision based techniques, a real-time mistakes' prevention can be performed, assuring a matching between real object features and virtual ones and allowing operator to immediately recover potential failures.

b) Automation issue When the component miniaturization increases, the capabilities of human hands to manipulate objects are no longer proper to guarantee the right tolerances, which obviously become smaller and smaller; moreover, microsystems are very susceptible to environmental conditions and they experiment counter-intuitive force

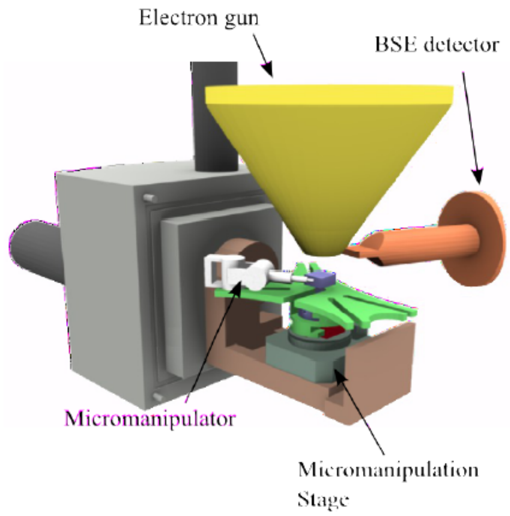


Figure 1.4: (Sauvet et al., 2009) propose a complete 3D modelisation of a SEM chamber

fields. For these reasons, the need for automation of micro-scale tasks arises. Human operators must be highly trained in employing powerful microscopes and tweezers, in order to succeed in often tedious and time-consuming tasks. In fields such as medicine and biology, the standard is still represented by purely manual manipulation, while in partially automated micro-manipulation stations, human dexterity and expertise should be transfered to the microrobotic station to ensure efficiency, precision and repeatability: tool's motion is a direct imitation of that of the human operator's hand.

Also **purely automated micromanipulation cells** exist, in which there is no direct connection between human hand and robot, as closed-loop control algorithms are set in order to fulfill the task. However, relying on the latter approach appears to be quite challenging, due to force scaling effects. In addition, the required supervision level during a microscale task is related to the structure's quality of the virtual environment, which could be affected by typical uncertainties in the microworld, such as limited working distance of microscopes, confined workspace, noisy visual sensing information or impredictable dynamic effects (such as friction and adhesion) (Ferreira et al., 2004). In certain industrial tasks, where environment is quite structured and events are mostly predictable, operation can be automated by computer-aided simulation techniques: however, as far as a non structured environment or a complex manipulation task are concerned, human supervision is necessary during unpredicted events or poor outcomes .

Moving from a manual to an automated platform with 3D virtual simulation increases at the same time performance and the **degree of intelligence** of a micromanipulation station. Main efforts in this scenario consist in disposing of a VR-3D reconstruction able

	Manual	Computer assisted
Vision	Eyes Cameras	Cameras
Assembly / Manipulation	Arms	Stages micro positioners
Tools	Tweezers	Grippers Automated tweezers Interfaces

Figure 1.5: Shifting from autonomous to assisted or teleoperated micro-scale tasks.

to provide human operators with an **immersive feeling** in the virtual scene, possibly integrated with other kinds of feedback, such as acoustic and force ones. In fact, the trasmission of skills and details from macro to microworld has to be as realistic as possible and depends also on the selected interface.
Relations between manual and computer-assisted microtasks in order to assure a realistic mimic of a highly skilled human operator are shown in fig. 1.5.

- Visually sensing positions of objects of interest, employing cameras and microscopes;
- Performing micropositioning using accurate and precise stages instead of directly moving with arms and hands the assembly area;
- Employing proper interfaces and tools to replace classical tweezers.

VR-based systems are able to properly modify human input, providing **reliability, safety and collision-free paths**: integrating a simulated microworld into a microsystem optimizes automation of tasks in the real environment.

c) Scaling issue It is not always possible nor convenient to adapt design and control of traditional manipulation/assembly solutions to the very strict specifications of the microworld.
If dimensions of manipulated objects are $< 100\ \mu$m, surface forces are stronger than gravity and, in general, than other volumetric forces. These forces are classified as:

- **van der Waals forces**, acting between the molecules of two near bodies and depending on materials in contact and on their distance;
- **electrostatic forces**, classic Coulomb forces that depend on charges present on surfaces;

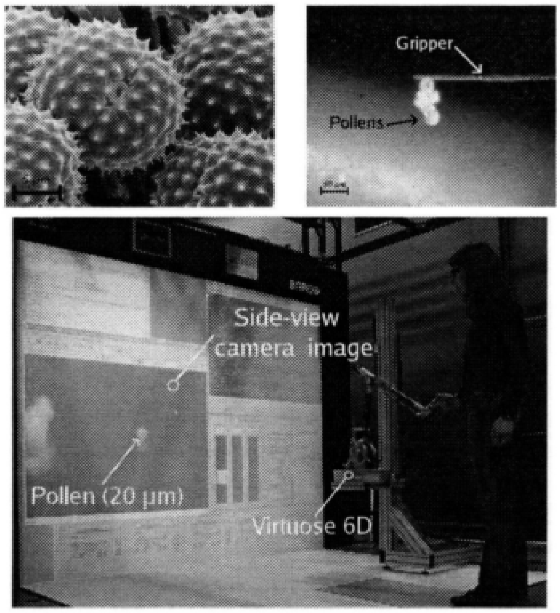


Figure 1.6: Ragween pollen manipulation performed with adhesion-based micro-manipulator (Haliyo et al. 2005)

- **capillary forces**, due to environmental humidity conditions, they act between two solids linked by a liquid bridge.

The impact of the scale change on the prevalent physics is a crucial issue known as **"scale effect"**: when miniaturization occurs, this effect modifies the strength of different physical phenomena, improving or deteriorating system's performances.
In fact, while predominant forces at the macro-scale depend on the object's volume (for a cubic object, they will be proportional to l^3), those which determine micro-objects behaviour are length-based (proportional to l). In the miniaturization process, consequently, the impact of volumetric forces decreases more rapidly than the effect of non-volumetric ones.
Manipulation by contact of objects between 1 μm and 1 mm is often disturbed by adhesion between the manipulated object and the gripper (Tam et al., 2009). In fact, a **specific modelisation** is needed in order to fully understand the physics at the microscale and to predict possible instabilities deriving from surface forces dominance. A fundamental characteristic to take into account while modeling is *roughness*, a surface topography factor which has been proved to influence manipulation: it causes contact to happen only at the asperity peaks and the contact area reduction leads to a decrease of electrostatic interaction. (Tam et al., 2009) have dealt with this aspect, conceiving a simulation tool based on finite elements to model surface roughness in order to provide valid and innovative design solutions for the conception of micromanipulators with controlled

adhesion.
A common approach to overcome the sticking phenomenon has been to use suction and vacuum instrumentation in order to minimize adhesion force. However, even performing an adhesion reduction by selecting materials with low van der Waals potentials or substrates with rough surfaces does not lead to an "adhesionless" behaviour. An original approach has been proposed in (Haliyo et al., 2003 and 2005, see fig. 1.6): in fact, adhesion forces do not represent only a limit for tasks at the microscale. They can be employed to carry out micromanipulation tasks, in terms of the only pick-up part of a complete pick-up and release task: this avoids designing complicated grippers' architectures, as the simple contact leads to task's success. Obviously, the release task becomes the most critical issue to solve at this point: a possible solution to reach the required force necessary to separate two medium in contact could be the use of inertial force derivating from an acceleration or to obtain the release by rolling, if a subtle control of the adhesion force is available. A one-finger gripper micromanipulator based on sticking and inertial effects (in other terms on adhesion and dynamics) has been developed, capable of performing both the sorting and the mechanical characterization of micro-objects.
The role of VR in four of the most interesting and widespread applications will be studied in the following sections, focusing on their relative issues and projectual choices to address them. Therefore, general-purpose micromanipulation, microassembly, surgery micromanipulation and bio-micromanipulation will be examined.

1.3.1 Micromanipulation

The task of grasping, moving, reorienting and repositioning micro-scale objects, commonly known as **micromanipulation** is essential to assembly and manufacturing microrobots. It can be performed with or without mechanical contact between the end-effector and the manipulated object: therefore, **contactless** and **contact** manipulation techniques can be distinguished. As far as a contactless task is concerned, it presents the advantage of not dealing with adhesion forces, but, on the other hand, gripping forces can not be controlled, as blocking forces applied to the object are weak. On the opposite, contact manipulation allows a controllable gripping force.
VR has been employed in (Abrahamians et al., 2013, see fig. 1.7 and Sauvet et al., 2012) in order to create a specific virtual environment for a real-time reproduction of a SEM chamber. In fact, performing micro and nano manipulations with SEM has proven to be a valid tool to explore these scales. Creating a 3D world which replicates SEM's experimental setup addresses classicaltypical SEM's issues, producing several advantages, such as:

- a clear and free view of robot motion's features is ensured, compensating the lack of natural visual access in SEM environment; moreover, operator's point of view in VR can change arbitrarily depending on tasks' specification and enhancing grasping

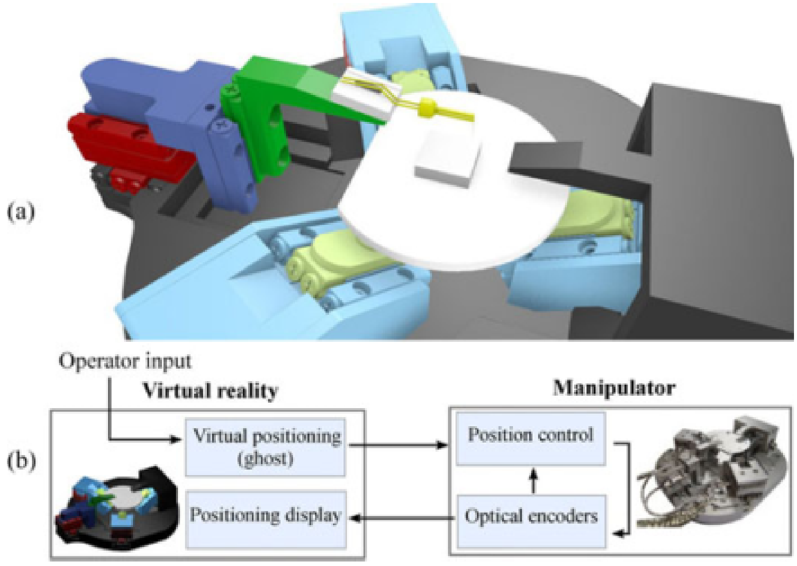


Figure 1.7: (a) Blender VR system interface (b) VR control scheme in (Abrahamians et al., 2013)

performances;

- implementing additional sensors is possible, in order to predict and avoid collisions between robotic structures, thus limiting procedure's risks, or to measure adhesion forces;
- while SEM manipulators usually present open-loop kinematics, VR allows the operator to act as a controller, so that the set-up can be considered as a closed-loop system.

A possible application has been explored in (Abrahamians et al., 2013, see fig. 1.7), achieving a mechanical characterisation of fragile resonant MEMS: in fact, a 9 DoF nanomanipulation system, working inside a SEM, has been coupled with a VR environment for analysis of local stiffness variations on suspended micromembranes through a self-sensing probe. In this case, a 3D Blender (professional free and open-source 3D computer graphics software) replica of the manipulation set-up has been realized, allowing both skilled and unskilled operators to simulate manipulations nearby the sample within the operating range with a free view of the area of interest. Using a 3D model permits, in this case, to extend the two-dimensional view provided by SEM.

Generally speaking, simulating and planning micromanipulation tasks on the virtual sample before performing them with the experimental set-up, avoids the risk of deteriorating the sample which, for example, may not be capable to sustain SEM's electron

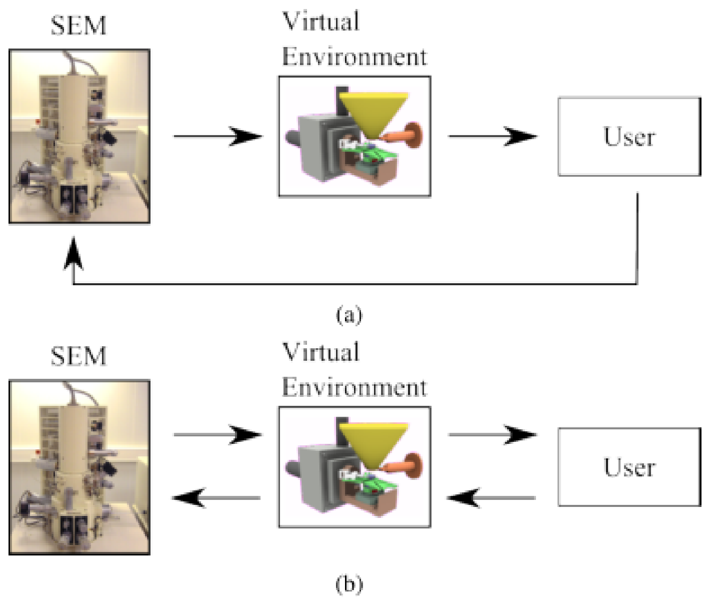


Figure 1.8: (Sauvet et al., 2013)'s system possible architectures. In (a) the open-loop one: user controls manipulation stage. In (b) the closed-loop one: user controls the virtual model.

beam for long periods. Such a step is also crucial for training inexperienced operators to deal with the micro- and nano-world and permitting them to safely manipulate the sample. Virtual environment presents itself as an intermediary between the real micro or nano-manipulation task and human operator. In fig. 1.8 two possible control schemes are represented: an open-loop one, in which the configuration of the virtual model in directly mapped on the real one, and a closed-loop one between SEM and Blender environment, where the virtual configuration is used as a set point for the real stage.

A reliable and accurate control is needed when micromanipulation is involved: automating tasks becomes essential to ensure repeatability, rapidity and safety. In this framework, **visual servoing** represents an unavoidable tool, as it provides accuracy in position and orientation of the micro-object of interest. This technique calls for controlling robot's motion using image data from one or more cameras. Two types of robot-camera relations exist: *eye-in-hand*, in which the camera is placed on the end-effector, and **eye-to-hand**, preferred in microrobotics, as the camera is fixed and points towards the end-effector. In (Cui et al., 2014) a control scheme has been implemented for visual servoing with photometric information for 6DoF: very little information is required as only image intensity has to be employed, so feature's extraction and motion's prediction are no longer necessary.

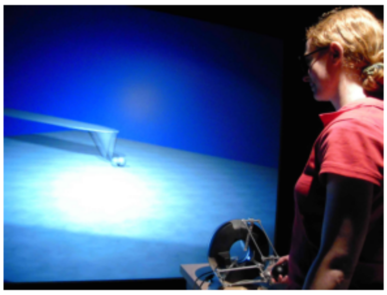
Teleoperation deals with the possibility of performing remote manipulation. Several

fields can take advantage of this technology: for example, risky interaction with wild environment can be handled and surgical procedures are optimized, thanks to tele-intervention of great surgeons who are able to operate remotely from large distances. Teleoperation also assumes a major role for researchers in micro- and -nano robotic field, in order to access specific equipments such as AFM cantilevers integrated into SEM. In this case, interfaces requirements are very specific and demanding: they should be as intuitive and natural as possible, assuring a clear and straighforward interaction between human end-user and the manipulated tool, even if non-expert users are involved.
Teleoperation fundamental specifications are the following:

- to achieve stability, computational time of vision tracking algorithms must be reduced: vision data represent the heaviest load on the system and in order to assure synchronicity between seeing and manipulation the amount of data transfer should be as low as possible, to reduce communication delay. For this reason, a convenient software architecture has to be found;
- accuracy;
- reliability;
- modularity to address different kind of applications.

Various strategies have been implemented to guarantee intuitiveness: in (Bolopion et al., 2011) virtual reality and haptic feedback are both explored in a teleoperation task (see fig. 1.9). The micromanipulation set-up (remote side) is settled up in Oldenburg (Germany) and it consists in an AFM cantilever integrated into a SEM, while the operator side, in Paris (France), includes the Omega haptic device to teleoperate the AFM's cantilever. Moreover, in Paris, operator can dispose of a stereoscopic 3D virtual reality scene (constructed using Blender software): it replaces sending over the complete amount of camera data relative to the cantilever's and to the nearest object's positions, in order to enhance system's stability and intuitiveness. As the cantilever only provides force measurements on the vertical direction, vision detection and robust vision tracking algorithm are required to provide additional force information about the manipulation task. So haptic feedback is enhanced thanks to visual one, which integrates information obtained through cantilever's force measurements: user's immersion sensation in the virtual scene increases.

Therefore, a further resource to provide users with the chance to experiment a more realistic feeling while manipulating objects is **haptic feedback**: experiencing a force feedback helps to extract useful information about the scene (Bolopion et al., 2010 and 2012). For manipulation tasks, haptic feedback provides a convenient and faithful rendering of microscale's interactions, along with a valid assistance to the operator for improved dexterity and collision avoidance.



(a) Teleoperation of microspheres using a haptic interface with a 3D virtual reconstruction of the manipulation scene

(b) Teleoperation between Paris, France and Oldenburg, Germany (630 km point-to-point distance)

Figure 1.9: (Bolopion et al., 2011) remote miscoscale teleoperation

The coupling between visual and haptic feedback represents a valid approach to enhance precision, repeatability and safety of tasks at the microscale. A further step forward is represented by (Amni et al., 2007), who propose a multimodal approach to perform telemicromanipulation with vision, haptic and sound feedback: system's architecture consists in a multisensory Human-Machine Interface (HMI) connected to an AFM-based micromanipulator coupled with an optical microscope. The contribution of VR, in this case, consists in providing haptic virtual fixture guides, which limit user's range of motion in freehand positioning, reducing their cognitive load: studies have proved that performances on a given task can be increased as much as 70 % after the introduction of virtual fixtures, as they increase operator's **vision** and **perception**. Moreover, safety is also ensured, as a decoupling between master and slave is guaranteed: the operator does not act directly on the microscene, but only on its virtual reconstruction. Experimental results show that this teleoperation scheme enables the operator to transmit human skills at the microscale, improving performances and reducing execution times.

1.3.2 Microassembly

The assembly process, already asserted at the macroscale during manufacturing, has been adopted in the very last years also at the microscale. Microassembly lies between conventional (macro-scale) assembly (with part dimensions > 1 mm) and the emerging field of nanoassembly (with part dimensions in the molecular scale, ie, < 100 nm).
Micro Devices Assembly (MDA) can be performed both manually and automatically; any microassembly task is generally associated with several issues:

- assembling a whole system composed by different microcomponents can be challenging, as materials could be incompatible with each other and several processes could be necessary to achieve final monolithic design: **flexibility** is indeed one of the ideal characteristic of a microassembly workstation, in order to produce hybrid, high-complexity, micro-scale devices;
- adapting traditional manipulation set-ups to perform at micro-scale presents several drawbacks: macro-scale robots show limited accuracy, due to inertia and mass-related robot characteristics, whose influence is minimized at micro-scale. On the other hand, disturbances such as small fabrication defects, friction, thermal expansion become crucial and **accuracy** requirements are in nanometer range

In (Fatikow et al., 1998), a flexible standard manipulation platform capable of real-time behaviour and of microassembly planning is proposed: it takes advantage of visual feedback provided by two sensors stations, one global (composed by laser measuring unit and a CCD camera) and one local (equipped with optical microscope and another CCD camera), which recovers manipulators and micro-objects positions. Assembly task planning is aided by simulation tool: CAD models of microrobots provide a task evaluation to select the best strategy in an early stage before the real operation. However, simulating a specific task is a challenge, as it is necessary to know forces occurring and consequently to discern dynamic behaviour of micro-objects.
CAD-CAM models, if fully available, are also used to reconstruct VR realistic environments: in (Ferreira et al., 2004) this VR technique, coupled with visual servoing, has been developed to carry out telemicroassembly operation, obtaining reliable position/-force feedback during automatic assembling of complex, hybrid MEMS. In this case, vision-based force sensing approach compensates the typical lack of force sensing in a conventional teloperated microtask: it is based on contact forces' estimation of the handled object through vision sensing of manipulator's tip deflection, during a common pick-and-place micromanipulation task. The microhandling workstation shows the following configuration (shown in figure 1.10): information flows between robot simulation system (the virtual microworld, which holds geometrical and kinematic data, user's interface) and assembly workstation (optical microscope, micromanipulators), linked by visual force/position feedback and by controller modules.

Force feedback represents a valid instrument in automated microassembly, a further resourse to maximize operators' performances: in (Bolopion et al., 2010), in particular, a dual gripping manipulation strategy is designed, controlled by haptic feedback, capable to realize microassembly of spherical objects (microspheres with diameter: 4-6 μm). As the microgripper is obtained from two AFM cantilevers, their deflection's amplitude lies at the basis of haptic feedback, which plays a double role, each one corresponding to a separate phase of the task's protocol. In first instance, during dynamic mode, in which operator the is able to freely explore the scene, changing nanostage and one cantilever's tip

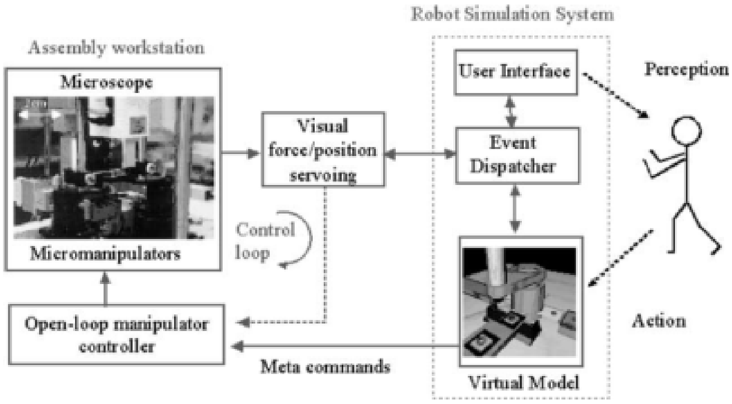


Figure 1.10: (Ferreira et al., 2004) teleoperated microassembly workcell under a microscope. Arrows represent the information flow.

positions through haptic interface, haptic feedback transmits to the user interactions at microscale. In a second phase, it is employed to enhance user's assistance, keeping his/her motion on a specific path: data recorded online during the first step are used to generate virtual guides to pull the user towards the optimal contact point with the microsphere.

1.3.3 Surgical micromanipulation

Microsurgery is a highly specialized technical discipline in Surgical Clinics: robotics can have a deep impact on it, helping surgeons to perform more precise and safer operations, or even to pioneer previously impossible procedures.
In fact, very specific sutures and repairs of nerves, vessels, but also eyes and ears, are involved, as well as neurosurgey and plastic reconstructive technques.
Microsurgery dimension involves a **transition zone**, between the limit of human vision and those of the optical microscope, in which very high precision, accuracy and dexterity are expected from the surgeon: 10 μm accuracy is often required.
For this need of extreme precision and accuracy, VR represents a useful tool to improve surgeons' performances, consequently reducing damaging risks for patients. This need for a very high positioning accuracy during micro-operations leads to take care of involuntary and inadvertent components in human hand movements, such as jerk, low frequency drift and physiological tremor, but also of involuntary hand movements with amplitude over 100 μm.
Physiological tremor has been defined as a roughly sinusoidal component inherent in human involountary motion, occurring at 8-12 Hz during vitreoretinal surgery, which can seriously affect surgeon's performance, risking serious collateral sight-damages for the

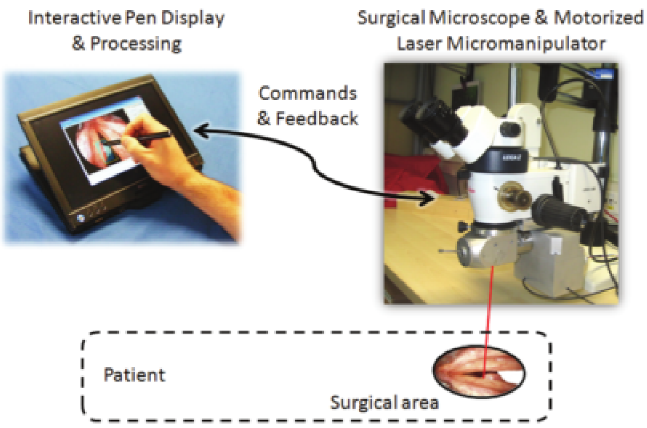


Figure 1.11: The "virtual scalpel" concept: the graphics pen controls in real-time the aiming and the activation of the surgical laser by touching the live microscope video shown on the interactive display

patient. Several handheld surgical tools and manipulators have been proposed in order to compensate this tremor (Rivière 2003, Latt et al., 2012), whose role becomes crucial in microsurgery tasks, as effective tremor suppression would result in greater precision, smaller incisions and better surgical outcomes in general.
Virtual fixtures (so virtual reality) strict guide motions in a selected direction. They have been employed in vitreoretinal surgery, requiring extraordinay precise micromanipulation, as the eye includes veins less than 100 μm in diameter and membranes only several μm thick. Employing virtual fixtures means to work on a very specific motion or task: in (Becker et al., 2013), *Micron* (see fig. 1.11) instrument is not directly manipulated by the operator through the application of forces, but only senses position in high bandwidth. The creation of virtual fixtures purely depends on the motion of the instrument's handle: vision, in this way, is employed to generate real-time fixtures, thanks to stereo-camera attached to the microscope. Surgeon is able to follow visual clues displayed and to keep hand-eye coordination.

Another interesting use of VR for microsurgery's tasks is presented in (Mattos et al., 2011, see fig. 1.11), where another tipical application of this field is taken into account: employing laser to cut or ablate tissues. The so-called "virtual scalpel" has been developed, performing laser tele-microsurgery from a computer monitor, using a graphics pen directly on the video taken with the OR's optical microscope. Experimental validation has shown how this technique can be intuitive if compared to a classical Microscope-Micromanipulator (MM) set-up: no training phase is necessary for users and an error reduction of almost 50 % when using the scalpel has been demonstrated. Moreover, this procedure addresses and solves the typical **hand-eye coordination issue** in microsurgery, improving, at the same time, ergonomics, controllability and safety.

Therefore, a new software for safety has been implemented, thus allowing the user to define safe manipulation zones, automatically switching off the laser if a risky region is detected.

1.3.4 Bio-micromanipulation

Bio-micromanipulation involves disciplines ranging from biology to engineering and includes operations as positioning, grasping and injecting materials into cells (Ouyang et al., 2007). It is possible to distinguish, as in micromanipulation, non-contact (such as laser trapping, electrorotation, ultrasound and nanovector-based delivery) and contact, referred to as mechanical micromanipulation, techniques (Ghanbari, 2012). This last solution is more desirable if compared to non-contact methods, as it allows to reduce serious cell damages provoqued by laser beams or to solve the lack of holding mechanism related to electro-rotation.
Analyzing samples at a single cell level represents a valid alternative to fully understand the fundamental elements of biological systems, if compared to classical analysis of average properties over a cell population (Yu et al., 2001), but, at the same time, it is quite a hard task.
Robotic technologies can help in order to improve accuracy and feasability of this kind of studies: in fact, microsystems are suitable for such applications as they're able to operate at the same scale of the organic cells, achieving in vitro interactions and reliable cell moving and sorting operations.
Manipulating single cells rather than a whole population plays a crucial role in the fields of molecular biology and drug discovery, in order to permit:

- diagnosis of particular **deseases**;
- fully **understanding biological systems and processes**;
- carrying out **in vitro** tasks on lab-on-chips in order to push and/or sort biological cells or to give a contribution to in vitro fertilization (ICSI: Intracytoplasmatic sperm injection);
- **improving technologies**: possibility to realize cell-based sensors;
- **research** interests in various areas: gene identification, therapeutic and regenerative medicine.

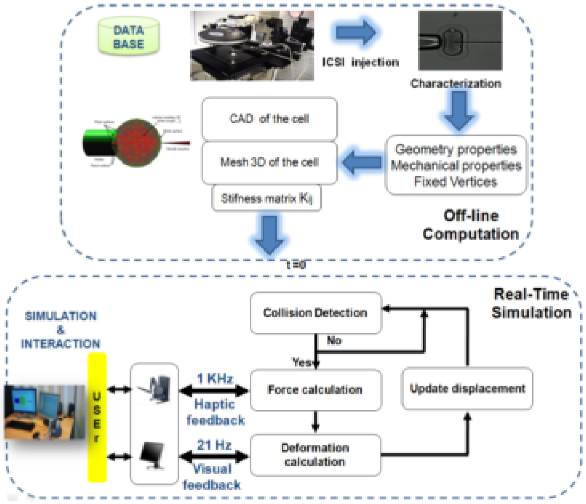
As great interests are involved, extremely complex and precise manipulation techniques are required: different studies have been developed about the role of **visual servoing** and **computer vision** in improving bio-micromanipulation tasks. In fact, bio-micromanipulation presents specifications which can be reached taking advantage of these tool:

- precise and accurate positioning;
- fast micro-object recognition to set the target injection point;
- fast image processing techniques to provide precise and online information to the control system.

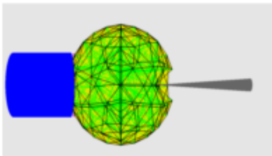
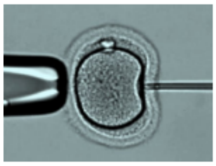
Moreover, the need for providing autonomous and controllable solution to bio-micromanipulation tasks has become urgent in order to overcome all the drawbacks related to atraditional manually-conducted cell injection (Ghanbari et al., 2009, Ladjal et al., 2011 and Ouyang et al., 2007).
Human performances are not reproducible and contamination is often introduced in this kind of techniques (inducing low operational speed and poor precision). Moreover, bio-operator can only count on his/her visual sense, without the possibility of adequately perceiving world at the microscale. Consequently, low (around 10 - 15 %) and strongly dependent on the operators' experience success rates are obtained, despite that training a bio-operator in this field is time-consuming (around one year to be able to conduct cell injection). The automation of the process brings to a great increase in the success rate.
In this scenario, (Yu et al., 2001) presents a visually servoed robotic system which performs automated pronuclei DNA injection using an hybrid control scheme and obtains 100 % injection success rate. Visual servo control has been employed by (Ghanbari et al., 2007) too, with the aim of moving a micropipette tip to the targeted position and deposing materials to that very point autonomously. A motion settle time of 0.5 s and a 1 pixel accuracy have been obtained, proving the efficacy of this kind of control, used along with a proper image processing algorithm.
Visual servoing is undoubtedly fundamental to provide precision and accuracy to micro-injection process and to change the way in which biological cells are studied and manipulated. Its development has been allowed by the advent of fast and cheap digital imaging and computer vision technologies. Selecting a visual servo-control in biomicromanipulation gives rise to several challenges (Ouyang et al., 2007):

- recognition of various objects in the scene: the cell itself, the end-effector of manipulators, which is in most cases an extremely thin tip of an injection pipette;
- a 2D image obtained from vision tools must be employed to control the system;
- a visual-servo control system has to be designed.

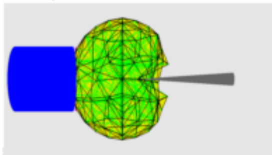
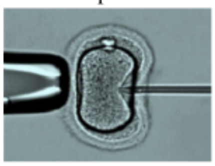
At the same time, it has been proved that assuring also an **haptic feedback** is a crucial issue depending on the knowledge of interacting forces experimented during the process. For this reasons, several bio-mechanical models of the needle-cell interaction have been proposed. Disposing of a force feedback supplies information about **needle penetration's speed and strength**, helping to control the whole procedure. Moreover, in



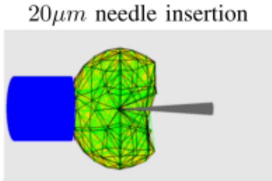
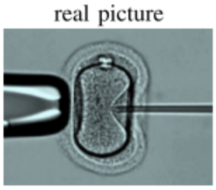
(a) Computational architecture for simulating force reflecting deformable cell micro-injection in a virtual environment.

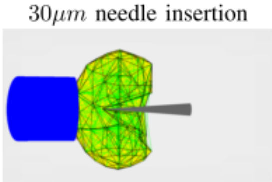
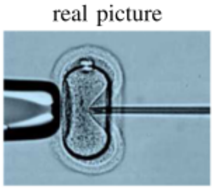
$10\mu m$ needle insertion real picture

$20\mu m$ needle insertion real picture

$30\mu m$ needle insertion real picture

$40\mu m$ needle insertion real picture

(b) Visual comparison of various stages of deformation during needle insertion between FEM simulations and experimental. data

Figure 1.12: (Ladjal et al., 2011) microrobotic simulator for assisted cell-injection.

order to minimize cell damage, it is important to reduce cellular deformation and to regulate needle's deflection. These specifications play a fundamental role especially for the development of microrobotic interactive simulators, with both bio-operator training and assistance functions. In fact, in (Ladjal et al., 2011, see fig. 1.12), a bio-mechanical finite element approach dedicated to real-time cell injection to help the training of biologist residents has been developed and implemented (see fig. 1.12). This simulator is able to provide both haptic and visual feedback, aiming to restore as accurately as possible bio-mechanical characteristics of needle-injection task, in a three-dimensional virtual environment.

The **role of simulation** becomes in this way prevalent, as far as biomicromanipulation is concerned: providing a real-time virtual reality scenario to the human operator allows his training on a three-dimensional and realistic cell environment, taking advantage of both sense of vision and touch, as the user is capable of actively manipulate cells if haptic interfaces are integrated too. However, building a 3D cell model is quite time-consuming and not easy to implement into real-time control applications: moreover, even if the model is in 3D, vision can only be 2D.

1.4 Robots for micromanipulation: parallel robots

Most existing robots nowadays present typical anthropomorphic characteristic, being clearly inspired to human arms: in fact, they are commonly known as **serial robots**, as they consist in a succession of rigid bodies, linked to each other by 1-Degree of Freedom (DoF) joints (Merlet 2001), which allow translation or rotation to one link relative to his predecessor.

A **generalized parallel manipulator**, on the other hand, can be defined as a closed-loop mechanism composed of an end-effector, generally a mobile platform, having n-Degrees of Freedom (DoF), linked by at least two independent kinematic chains, called *legs*, to a fixed base (Merlet 2008). Such a system represents a particular class of closed-loop kinematic chains, in which links and joints are arranged such that at least one closed loop exists. Moreover, this definition is very open, as other kinds of mechanisms, like redundant or cooperative ones, are included as well. A visual comparison between serial and parallel robots is provided in fig. 1.13.

1.4.1 Comparing Performaces of Serial and Parallel Robots

Resorting to a parallel architecture instead of a serial one can result convenient under mulitiple points of view, especially when a micro-scale task along with its respective scaling issue is involved. Despite that, parallel mechanisms also exhibit some drawbacks,

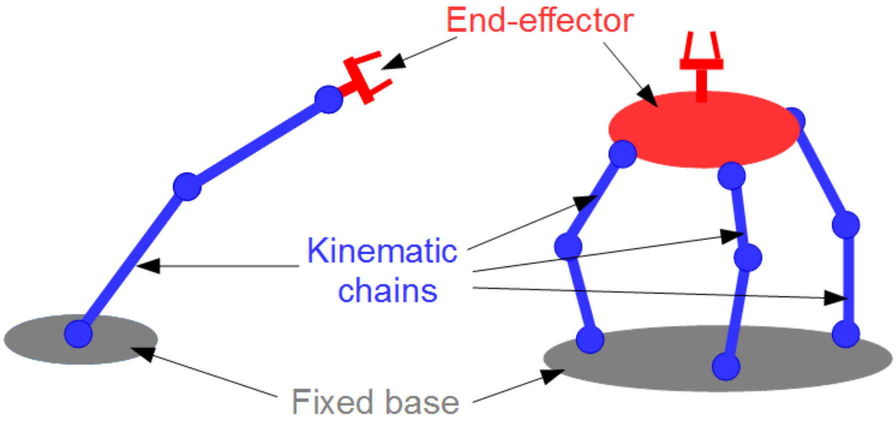


Figure 1.13: Schematic comparison between simplified serial (on the left) and parallel (on the right) manipulators, in terms of their constitutive elements: fixed base, kinematic chains, end-effector.

such as a small workspace, a larger number of mechanical components and a more complex study than serial robots. Theoretically speaking, shifting from a serial architecture to a parallel one means distributing the load on links: in fact, the end-effector is connected to the fixed base through a series of legs, which support only a fraction on the total load. In the following section, consequences of this architectural changes will be analized in details, studying several parameters of interest in the evaluation of a robotic mechanism's performances and making a comparison between serial and parallel manipulators.

Load-carrying capacity While performing a general-purpose task of moving an objects by the end-effector, each serial robot's actuator must rely on sufficient power to move not only the manipulated objects, but also links and actuators in between. In fact, it is sufficient to think about serial robot's architecture to understand the reason of this behaviour: the end-effector is located at the end of the kinematic chain and each link has to support the weight of the following segments and of the load itself. Links experience large flexure torques and must be stiff, thus becoming heavier. Consequently, serial manipulators generally present a poor payload/robot mass, typically smaller than 0.15 for serial 6R industrial robots. The same ratio, for parallel ones, can be larger than 10: the load, in this case, is equally distributed on all actuators and, as joints can only impose traction-compression constraints, flexure imposed on links is minimized. Consequently, it is possible to employ lighter and smaller links, decreasing the whole system's mass, and to select actuators with lower power.

Positioning accuracy The definition of positioning accuracy includes both absolute accuracy, meaning the distance between the real and the desired EE position and

repeatability, intended as the maximum distance between two EE's positions reached for the same desired final position, starting from different points. Positioning accuracy depends on several factors, such as: accuracy of system's internal sensors, links' flexure, quality of geometrical assumed constraints between axes of the links and friction.
Serial manipulators go through error's accumulation from one joint to the next one; moreover, the influence of a joint's error on the EE is larger when the joint is close to the robot base. Links, in this sense, magnify errors and a small measurement error provided by internal sensor will lead to a larger error in EE's positioning. Parallel robots, on the other hand, thanks to their architecture, are able to provide a significant rigidity, even with lighter links, along with a very good positioning accuracy. Moreover, when linear actuators are involved, this last feature increases as flexure-related deformations are minimized and, unlike what happens with serial manipulators, measurement errors in the internal robot sensors minimally influence the EE's positioning.

Scaling issue As far as serial manipulators are concerned, it is not possible to proceed successfully in a miniaturization process simply scaling down a macro-scale version. In fact, their behaviour is strongly influenced by inertia and friction: while the first force does depend on the links' lenght, the second one doesn't. As a consequence, scaling down dimension would result in a minimization of inertial forces, while friction would be unchanged. Parallel structures appear to be almost insensitive to scaling and any type of transmission can be employed in their design.

Dynamic behaviour Dynamics is a very important constraint to take into account, especially when execution speed is essential, such as in pick-and-place operations, where the so-called *fast robots* are greatly employed thanks to their light-weighted links. When an interaction robot-environment is required, such as in rehabilitation or medical robotics, dynamics plays an important role too. For what has been stated above, serial manipulator's links must be stiff in order to avoid excessive flexure: during a task execution these systems experiment inertia, centrifugal and Coriolis forces, which deteriorate their dynamic performace. Moreover, at high speed, such systems tend to vibrate. On the other hand, parallel robots possess a better dynamic behaviour than serial ones, thanks to the high payload over robot mass ratio, to the reduced coupling effects between joints and to link's light weight.

Workspace Usually, parallel robots exhibit a small workspace, if compared to serial ones. In particular, if the system foresees more than 3 DoF, workspace's shape is more complex for parallel robots and no graphical illustration of it will be possible. For serial robots having the same number of DoF, this problem does not arise.
Workspace issues arises for various reasons, such as the possibility of self-collisions occurring between robot's elements, or mechanical limitations related to passive joints, or,

in some cases, the leg's lenght ranging from a minimum to a maximum value depending on the linear actuator's motion. Moreover, within their workspace, singularities can appear, splitting the workspace into separate components: this risk, however, occurs also in serial robots.
In conclusion, calculating workspace for parallel robot is a hard task: even if the geometry is perfectly known, one must take into account that manufacturing errors may affect workspace's shape.

1.4.2 Parallel robot's architectures

In fact, in order to optimize parallel robot's performances, disposing at the same time on a relatively simple structure, several features are desirable:

- as stated in the first general definition, at least two legs connect fixed base and end effector and each of them includes at least one simple actuator;
- the number of actuators should be minimal, or, in other words, equal to the number of DoF of the moving platform. Otherwise, **redundancy** will occurr. This feature rather complicates the robotic system's modeling, even if can provide, at the same time, very good chances to deal with important and complex issues typical of parallel robots, such as avoiding singularities and solving direct kinematics. In general, a redundant robotic system also presents improved manoeuverability and an increased workspace, which can be particularly useful for parallel systems, as will be explained further on;
- the moving platform shouldn't have any type of mobility when actuators are locked: especially in the medical field this feature is essential to guarantee application's safety.

Depending on parallel robot's specific architecture and structure, on joints/links arrangements, various definitions and classifications have been provided in literature in order to try to establish some fixed laws to solve their kinematics and dynamics.
The first important difference to underline is the one between **fully parallel** and **not fully parallel** manipulators. When the robot's number of chains is strictly equal to the moving platforms' number of DoF, then the manipulator is called fully parallel.
Supposing that legs are identical and depending on their architecture and mobility, fully parallel systems are classified as follows:

Planar robots Their performances are carried out only on a plane: so their mobile platform envisages 3 DoF, two translations and one rotation aroung the normal to the place of mobile platform. An example of planar robot is shown in fig. 1.14.

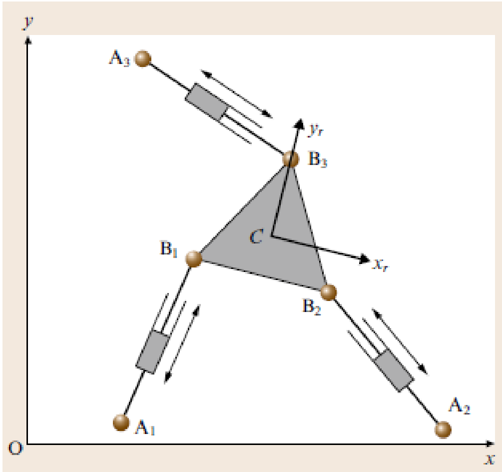


Figure 1.14: A representation of a 3-DoF RPR planar parallel manipulator, with its global and local reference frames

Spatial robots They have of 3, 4, 5 or 6 DoF, with mixed types of joints in each chains, and operate in the whole space. A very interesting and world-wide known family of parallel robot is the Delta (shown in fig. 1.15), first developed at Ecole Polytechnique of Lausanne (EPFL) by professor Clavel, with its 3 translational DoF plus a rotational one, typically employed in pick-and-place and machining operations. It is made up of three legs connected with universal joints to the base. The most important design feature is the employment of parallelograms in legs: in this way, the EE's orientation is maintained fixed, restricting its motion to pure translation. The Delta robot finds many application fields, thanks to its high speed, stiffness and high accuracy: it has been employed, in various commercial versions, in food packaging industry, surgery, assembly of MEMS, also to realize haptic interfaces (e.g. Force Dimensions) and, more recently, in 3D printers.)
When the robot possesses 3 rotational DoF it is known as orientation manipulation.

For 6-DoF spatial robots, when moving platform and fixed base are linked by 6 legs, generally driven by 6 prismatic actuators, Merlet has provided a commonly accepted classification, which depends on base's and platform's shapes and on the linking points' positions between them (see fig. 1.16):

- robot **MSSM** (Minimal Simplified Symmetric Manipulator), characterized by triangular base and mobile platform, while legs are mounted by pairs at both ends of the upper triangle's vertex;
- robot **TSSM** (Triangular Simplified Symmetric Manipulator), with hexagonal base and triangular platform, in which two legs are connected on the same triangle's

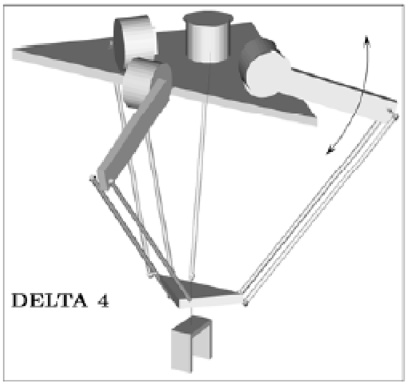


Figure 1.15: A Delta spatial robot schematic representation and one of its industrial versions, the FlexPicher IRB 340

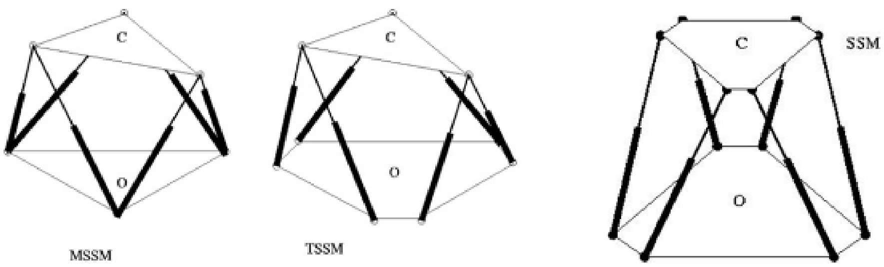


Figure 1.16: 6-DoF fully parallel robots Merlet's classification

vertex;

- robot **SSM** (Simplified Symmetric Manipulator), whose base and mobile platform are both hexagons, while legs are connected to the vertex of the hexagons.

In any case, depending on the type of joints employed in each chain, they are identified with joints' initial letters: for example, if a chain's configuration is composed by one Prismatic joint and two Rotational joints, the chain will be identified as a PRR chain.

Non-conventional parallel 6-DoF robots architectures exist as well, which do not follow strictly the classifications above. Two interesting examples are the following:

- **decoupled** robots, have been designed in order to simplify parallel robots' control: while in a standard parallel manipulator each actuator contributes to both positioning and orienting the moving platform, these mechanisms present three actuators which control its translational motion and the other three which control its orientation;

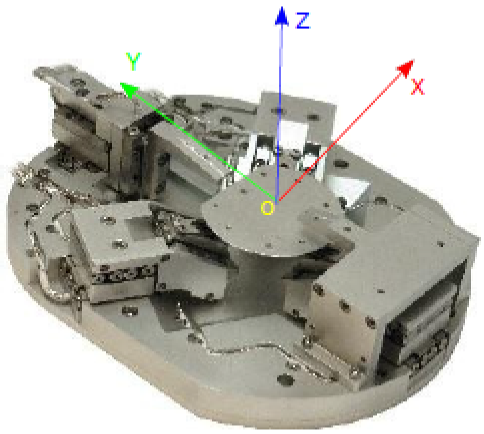


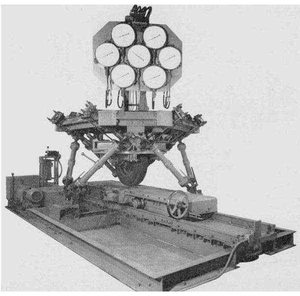
Figure 1.17: A commercial example of a 3-legged not fully parallel robot: SmarPod 70.42s robotic manipulation platform (SmarAct GmbH), with his local reference frame.

- **three-legged** robots, with two actuators per leg, fall outside fully parallel robot's cathegory, as the number of DoF of their end-effector is larger than the number of chains (see fig. 1.17). Various authors have proposed such architecture, because it helps to decrease the risk of interference between legs, leading to an increase in workspace dimensions. On the other hand, system's stiffness is reduced and positioning errors increase. This kind of structures, to guarantee motion transmission without becoming redundant, usally employ also passive joints: however, it has to be underlined that accuracy gets worse as the number of passive joints increases.

1.4.3 Applications

Since Gough in 1947 first explored the basic principles of a closed-loop kinematic chain, capable of positioning and orienting a moving platform, parallel robots have been successfully employed in many applications in which a high load carrying capacity, good dynamic performance and precise positioning are of paramount importance (Zhang et al., 2011). Gough's idea (see fig. 1.18) lied in an hexagonal platform, linked to a fixed base thanks to a ball-and-socket joint for each vertex. All the six legs had variable lengths, thanks to 6 linear actuators, one for each of them, which allowed modifying the position and the orientation of the moving platform. 6 universal joints were put at the end of each leg, to link them to the base.

Gough realized his first prototype in 1955 and, from 1965 on, Stewart started to popularize this kind of parallel mechanism, applying its working principle to flight simulators. Nowadays, range of application for parallel robot stretches from astronomy to medical



(a) Original Gough platform (1947)

(b) Last industrial prototype of Gough platform (2000) to be used in the Dunlop Tyres company

Figure 1.18: Evolution of Gough's platform

rehabilitation, from vibration damping to industrial devices requiring ultra-accurate positioning.

One of the first application fields for parallel systems has been the industial one, especially when concerning the achievement of a fine positioning device: in this sense, they were adequate to assembly and microassembly tasks. Many companies, such as PI (Physik Instruments), Micos, Alio, etc. show a wide variety of positioning systems, such as hexapods.

Focusing on the medical field, MIS has taken advantage of parallel mechanism's property to be almost insensitive to scaling, confirming a trend towards miniaturization. These small and adaptive systems have been employed, for example, for complex and delicate ophtalmological surgery operations. The INRIA active wrist (Merlet, 2008) has been designed to address typical surgical issues, such as reliability, bio-compatibility, minimal size, ergonomy. Another interesting system is the ISIS SurgiScope®, a robotized tool-holder employed in microscope-assisted surgical neuronavigation. This commercial device is fully operated by surgeon thanks to an user-friendly touch screen and its image injection module displays data directly within the surgeon's field of view.

Chapter 2

MODELING A COMMERCIAL PARALLEL MICROPOSITIONER

2.1 Thesis objectives

The main **purpose** of this project is the development of a commercial micropositioner (SmarPod 115.25, SmarAct GmbH, see fig. 2.2(a)) for a Geometrical Model (GM). SmarPod is characterized by **parallel kinematics** and is employed, being vacuum-compatible, for precise and accurate sample's positioning under SEM, for various applications.
Geometrical modeling represents the preliminary step to fully understand, and possibly improve, robot's closed loop behaviour in terms of task's quality, especially when manufacturers do not provide sufficient documentation. SmarPod micropositioner is employed to perform very precise micropositioning tasks: the respect of **accuracy** specifications and requirements is essential, especially when **visual servoing** (Cui et al., 2014) or measuring mechanical properties of delicate microstructures (Abrahamians et al., 2013) are concerned. The studied robotic system represents a *"black box"*, from which it's possible to extract useful information. This step is crucial in order to improve the reliability of bio-microsystem's manipulation and characterization.
Disposing of a detailed microrobot's model becomes essential, in addition, (SmarPod's CAD model is shown in fig. 2.2(b)) to deal with the typical lack of sensing at the microscale, as it allows a 3D precise and adequate reconstruction, realized through proper softwares, of the manipulation set-up.
The roles of **Virtual Reality (VR)** and of simulations, carried out in Blender (*Blender foundation*) environment, are asserted as essential tools in microsystem's task planning. Blender is a professional free and open-source 3D computer graphics software and it is proven to be a basic instrument to validate microrobot's model, also to simplify it in

case of complex system's geometries.
To obtain an adequate model of the micropositioner, a step by step **workflow** (see fig. 2.1) is followed: in the first instance, SmarPod's positioning system is studied in all its technical features, to fully understand its architecture before starting modeling. A definition of Direct Geometrical Model (DGM) and of Inverse Geometrical Model (IGM) is then provided. Both SmarPod's moving platform's positioning and orientation are considered, therefore a first solution for the IGM is developed, called **Analytical IGM** (A-IGM).
Then, possibilities supplied by Blender open-source software are explored, exploiting a SmarPod's existing model loaded in the virtual environment (see fig. 2.2(c)). Active joint variables' values and other interesting features are extracted, thanks to simple movements imposed to the virtual robot: to obtain them, the development of a **Python code** in Blender environment is necessary.
Taking advantage of the observation of basic movement's sequences performed in the VR environment, the so-called **Simulation-Aided** SmarPod's geometrical model is conceived (SA-IGM): performing similar motions on the real robot, on such a smaller scale, wouldn't have led to the same conclusions, as it would have been impossible to observe them in such a **detailed and controlled** manner.
Joint variables' values obtained with the first Analytical IGM and the second Simulation-Aided IGM are compared to joint variables and other useful features extracted from Blender's environment; the most precise and accurate model between the two provided will be finally validated experimentally in the ISIR's clean room, on the real SmarPod, through the employment of a **microrobotic workstation**.

2.2 SmarPod micropositioning system

2.2.1 System's overview

This thesis is focused on the geometrical modeling of a commercial robot: SmarAct GmbH sells on market SmarPod micropositioners, characterized by parallel kinematics and employed for applications requiring high precision and accuracy, especially, being vacuum-compatible, for micro-positioning of samples in SEM environment with the final aim of performing a characterization of micro-biosystems.
Micropositioner's **main features** are its compact design, high stiffness and repeatability (see tab. 2.1), along with the possibility of defining its virtual pivot via software and to obtain large travel ranges. In addition, the system is HV and UHV compatible, realized with non-magnetic, UV-resilient materials.

SmarPod 115.25 micropositioner (dimensions 115x101.4x25 mm^3, see fig. 2.3 for the global reference frame chosen by SmarAct for all SmarPod's) weights about 400g and its

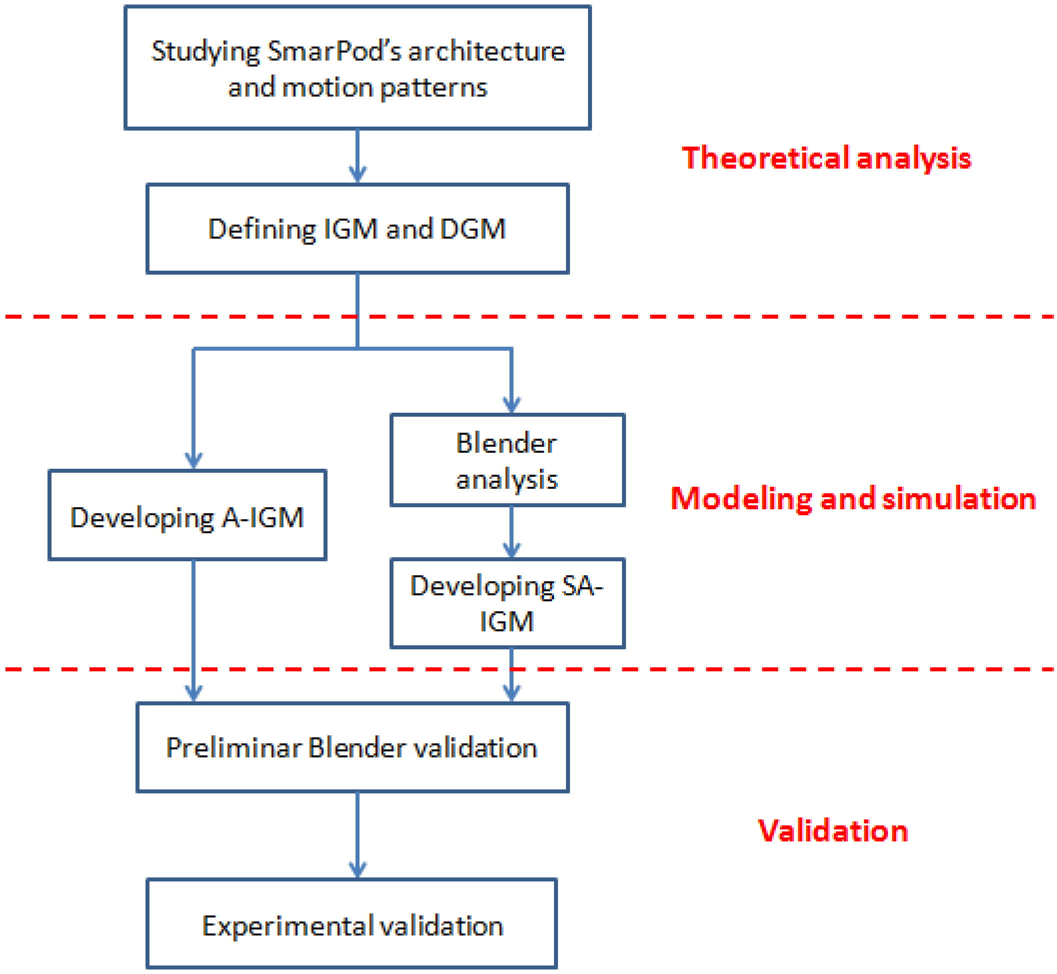
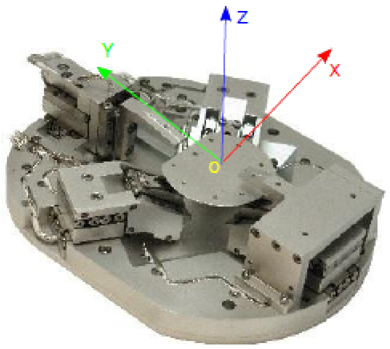


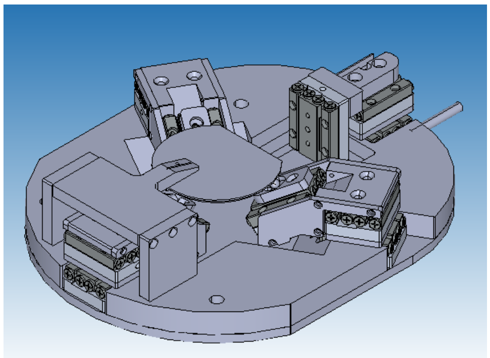
Figure 2.1: Step by step workflow of this thesis.

Smallest Increment:	Repeatability:
X, Y, Z: 1 [nm]	X, Y, Z: 200 [nm]
θx, θy, θz: 1 μrad	θx, θy, θz: 10 [μ rad]

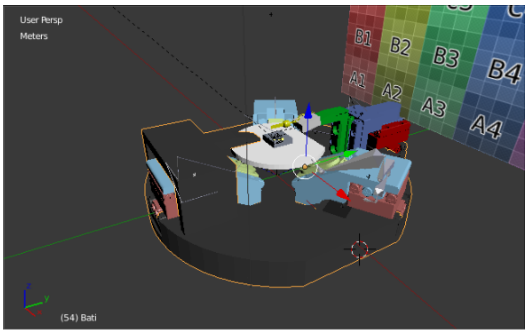
Table 2.1: SmarPod's 115.25 closed-loop repeatability with full-travel range: much better repeatability are obtained for smaller ranges.



(a) SmarPod 115.25 commercial micropositioning system by SmarAct GmbH (dimensions 11x13x4 cm)



(b) SmarPod's CAD model



(c) SmarPod's VR model on Blender

Figure 2.2: Shifting from real robot to virtual one

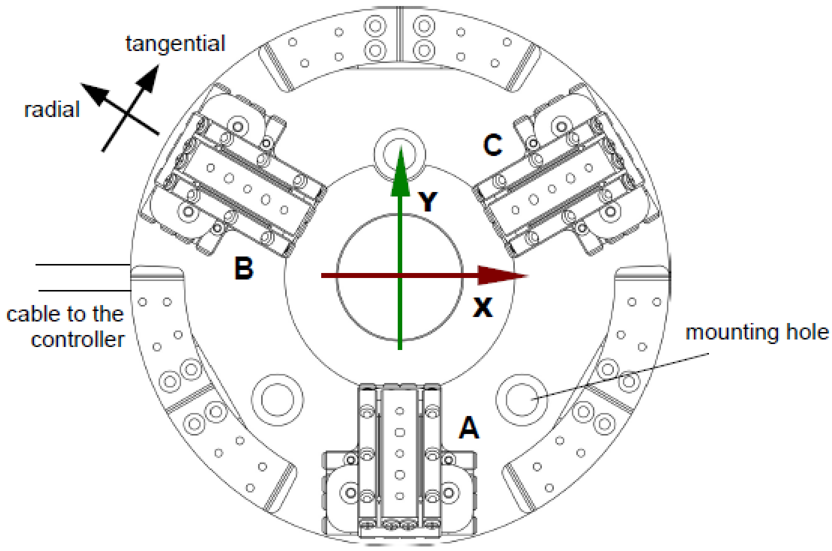


Figure 2.3: A rotational SmarPod, seen from above with its coordinate system: only positioners of the first two stages are shown.

travel ranges, both linear and angular, are shown in tab. 4.2.

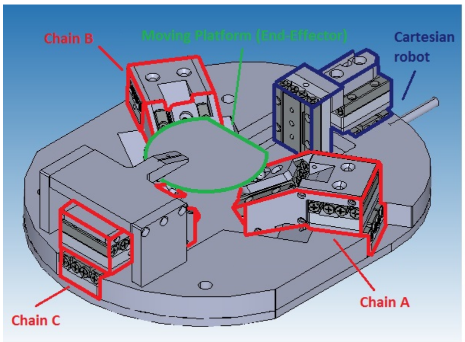
X: 10 [mm]	θx: 17 °
Y: 10 [mm]	θy: 20 °
Z: 5 [mm]	θz: 35 °

Table 2.2: SmarPod's 115.25 full-travel ranges

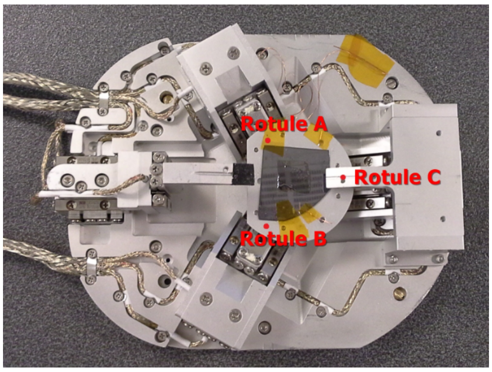
2.2.2 Micropositioner's architecture

This micropositioner is composed by two subsystems (see fig. **??**): a 3-axis nanomanipulator, in other terms a classical Cartesian robot, and a parallel robot: this thesis will focus only on the latter. It consists of three identical kinematic chains (or legs, named A, B and C): each of them includes three stages, which connect the fixed base to the moving platform, the so-called hexapod platform, the system's End-Effector (EE) and sample holder. Like most parallel robots, this system presents many unactuated (or passive) joints.
For each chain, the kinematic configuration is identical and consists in (see fig. 2.5):

- the **first stage**, the one directly fixed on the base, is provided with three linear actuators oriented radially to the base's center and called A-radial, B-radial and



(a) SmarPod's 115.25 subsystems: cartesian nanopositioner in blu, parallel robot in red and hexapod moving platform in green.



(b) Positions of rotules are shown on the real SmarPod.

Figure 2.4: Overview on SmarPod's architecture.

C-radial;

- the **second stage**, the middle one, is provided with three linear actuators as well, but all perpendicular to the radial ones and called A-tangential, B-tangential, C-tangential;
- the **third stage** (not shown in fig. 2.3, see fig. 2.4), the upper one, consisting of three passive guides which transmit first two stage's motion to the moving platform, thanks to three passive spherical joints (or rotules).

Actuators (SmarAct linear nanopositioners SLC-1720-S, whose dimensions are 22x17x8.5 mm^3) are all identical and **Stick-Slip** (SS) based. Positioners' mechanical and drive properties are listed in table 2.3.

Positioners from the SLC line of SmarAct are based on linear crossed-roller slides, which

Weight:	about 13 [g]
Travel:	12 [mm]
Step width:	50 to 1500 [nm]
Scan range:	1.5 [μm]
Scan resolution:	sub-nm
Velocity:	13 [mm/s]
Max. frequency:	18.5 [kHz]

Table 2.3: SLC-1720-S positioners mechanical and drive properties

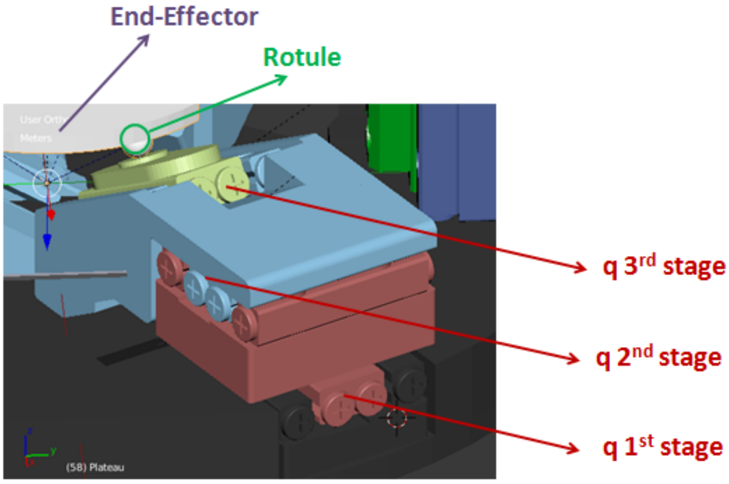


Figure 2.5: Single chain structure

confer them high rigidity and straightness. An inertial drive locomotion principle is employed, in other terms a Stick-Slip driving principle. As already stated in Chapter 1, piezoelectric stick-slip actuators are widely used in micro and nanorobotic fields, due to their high resolution positioning. Systems employing SS actuators are often dedicated, like in the present case, to automatic positioning in typical vacuum environments, such as the SEM. SS actuators are able to work in two different modes, depending on the applied voltage (see fig. 2.6): with the **step mode** a sub-nanometer resolution (single step's width down to 50 nm) is obtained, while the maximum travel range is achieved with the **scan mode**.

- In the **step mode**, the piezo gradually stretches with the slide, thanks to the relative friction between them, as a slow increasing voltage is applied: in this way, high step resolution is achieved.

- With the **scan mode**, a theoretically unlimited displacement (actually limited by slide's dimensions) is obtained. Voltage is quickly restored to zero, causing the piezo to suddenly shrink, while the moving mass slides on it, as the inertia force has become stronger than the relative friction. This process results in a forward net slide's displacement. Repeating this procedures results in a longer step size.

SLC actuators present several other good features, if compared to traditional positioners, such as: good miniaturization, backlash-free positioning, thanks to the absence of gears inserted, and crash-tolerancy, as the positioners are not damaged if moved against an obstacle or to a mechanical stop.

The suffix "-S" added to the positioner's code indicates the presence of an integrated sensor for nanopositioning tasks: an optical encoder, with a resolution of about 1 nm

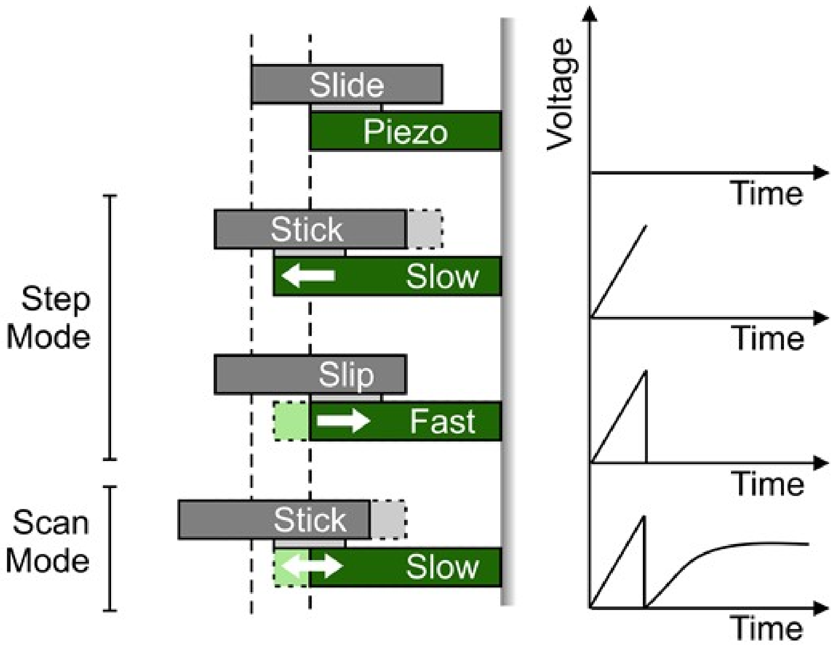


Figure 2.6: Stick-slip double phase (step mode and scan mode) driving principle

and an absolute accuracy of +/- 1 μm. which does not affect positioner's dimension. Through the employment of Modular System Control (MCS), SmarPod's control box, it's possible to process data extracted from nanosensors for closed-loop position control.

2.2.3 Calculating the number of Degrees of Freedom (DoF)

Before starting with GM's definition, an important parameter of the system has to be calculated: the number of Degrees of Freedom (DoF). Only after having obtained this value, it will be possible to go ahead with identifying system's inputs and outputs.
For parallel manipulator's determination of DoF, the following general relation is usually employed (Khalil et al., 2004):

$$N = \sum_{i=1}^{L} m_i - \sum_{j=1}^{B} c_j$$

where:

- N gives the number of DoF of the mobile platform, respect to the fixed base;
- L is the number of joints;
- m_i is the mobility of $i-th$ joint;

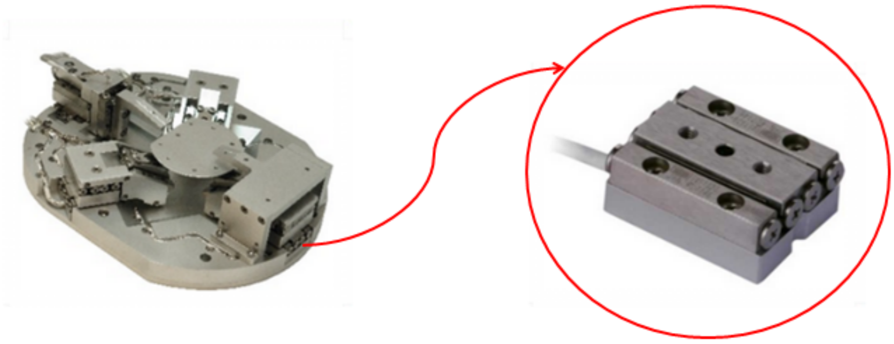


Figure 2.7: The robotic micropositioner and its linear actuators, both provided by SmarAct GmbH.

- B is the number of independent closed loops equal to number of chains $(n_c) - 1$
- c_j is the number of constraints of the $j - th$ loop (6 for a spatial loop and 3 for a planar one).

If kinematics chains between base and platform are **identical** and if kinematic loops have the same number of constraints, the previous equation becomes:

$$N = (n_c \times d) - (c_j \times B)$$

in which:

- d is the sum of DoF of all joints in a single kinematic chain.

SmarPod's micropositioner respects the specifications of the last formula. For it, resulting parameters are the following:

- $n_c = 3$: it possesses three groups of positioners or legs;
- d = 6. Each chain presents: 2 linear actuators (1 linear actuator = 1 DoF x 2 = 2 DoF), a passive slide (1 passive slide = 1 DoF) and a passive spherical joint as a linking point between the leg itself and the moving platform (1 spherical joint = 3 DoF);
- $c_j = 6$

- B = 2 (B = n_c - 1);

So the formula becomes:

$$N = (3 \times 6) - (2 \times 6) = 6$$

With 6 DoF in its moving platform and only 3 legs, SmarPod is included in the cathegory of **not fully parallel manipulators**. In fact, in literature, various 6-DoF robots with only three legs have been proposed, having two actuators per leg, like SmarPod's positioning system. The main reason to select such a configuration lies in legs' interefer-ence minimization, which consequently increases workspace's dimensions. However, an important drawback is involved too: system's stiffness results reduced, while positioning errors increase.
Moreover, accuracy gradually collapses as much as more passive joints are involved in the structure (Merlet, 2001). In this very case, the whole architecture includes two passive joints for each chain: the third's stage passive slide and the passive spherical joint which links it to the EE. Consequently, in SmarPod the number of active joints is equal to the number of DoF of its EE: it is a **non redundant** parallel micropositioner.

2.3 Defining IGM and DGM for parallel robots

GM to calibrate serial robots are well known in literature and fully understood, but equivalent ones for parallel mechanisms are relatively few and, above all, they're focused on *fully parallel* parallel robots and not on *not fully parallel* ones.
Inverse Geometrical Model (IGM, or Inverse Kinematics IK) is one of the basic elements of any robot controller. It consists in establishing the actuated joint variables' values (collected in a $\overrightarrow{q}$ vector) corresponding to a certain EE's pose. The most traditional way to represent a pose of a rigid body is to use coordinates in a selected reference frame of a given point of the rigid body itself (in this case, the central point E of the moving platform will be the reference point), along with three angles for the platform's orientation. Position and orientation of the EE are collected in a 6x1 $\overrightarrow{x}$ vector.
Direct Geometrical Model (DGM or Forward Kinematics FK) deals with finding the possible pose of a parallel robot's platform from the minimal set of the actuated joint coordinates. It is useful for velocity control of the EE and for calibration and motion planning tasks.
IGM for parallel robots is quite straightforward to solve, as there's a unique solution of the IK: this means that each joint variable can be computed independently once the final pose of the EE is given. On the other hand, DGM is very difficult to derive: usually, a non linear set of equation admitting multiple solutions is found (for example, Gough platform can have up to 40 solutions (Nelson et al., 2008)). The problem's solution is not unique : there are several ways of assembling a parallel manipulator with a given

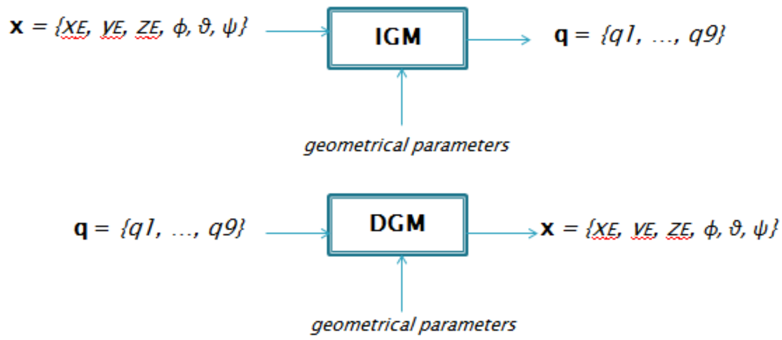


Figure 2.8: IGM's and DGM's formulation for SmarPod's.

joints' configuration. In literature, various solutions to this issue have been proposed, for example elimination, Grobner bases and interval analysis. However, employed algorithms involve long computational times, which hamper a real-time positioning control of the device. Furthermore, there is no known algorithm which allows to choose among the set of solution obtained for the current platform's pose.
In fact, in SmarPod's positioning system, it results impossible to impose through its PC interface joint variables' values, thus obtaining a certain final pose of the moving platform. On the contrary, one can select a certain final position of EE's central point E, but, even in this case, joint variables' values to reach it would not be available from SmarPod's Modular Control System (MCS) controller.

2.4 Analytical Model

In order to find equations to solve IGM, an assumption has to be made, in the first istance, concerning the system's **linearity**: it's therefore possible to apply the **superposition principle**, valid for all linear systems. In this way, the problem results simplified and split into two parts, position and orientation: they will be both analyzed in the following sections.
In the model, both actuated and unactuated leg's joint variables will be considered, in order to try to simplify the model's resolution; it is useful to obtain also passive slide values as they transmit movement to the EE. However, DoF relative to passive spherical joints will not be considered (see section 1.4.1 for further explainations).

2.4.1 Position

As far as EE's **positioning** is concerned, the main idea is to treat each leg like a **simple serial kinematic chain**, imaging to remove the moving platform, cutting in correspondance of the three passive spherical joints. A, B and C points' (see fig. 2.4(b)) coordinates

will be expressed as functions of both the joint variables of their relative leg and of their position in the platform, relative to the central reference point E.
These relations will be linked to each other by applying **closure equations**, restoring micropositioners' initial parallel structure. This means expressing the position of the i_{th} spherical joint in this way:

$$p_i(q_i) = p_i(x)$$

Considering for example rotule C, in other terms the spherical passive joint linking leg C to the EE, it can be treated both like being part of the leg or of the platform. This approach can be extended to the other two legs, as SmarPod's kinematic chains are all identical between each other.
A similar method has been employed by (Bicchi et al., 2008) to solve the IK of a fully parallel manipulator, but it could be extended also to this specific not fully parallel case. As authors explain, it is convenient to operate *"cuts"* between legs and moving platform directly in correspondance of the linking points between them: in this way, the $\overrightarrow{q}$ vector does not contain variables expressing EE's pose relative to the last leg of the chain. Consequenlty, in SmarPod's case, it will not contain joint variables relative to the spherical joint itself. Equations, in this way, will result easier, as they contain less variables.

In this purpose, gradual steps to achieve final equations will be now illustrated (for further details and complete calculations see Appendix):

- **Step 0:** Global and Local reference frames are selected. If n is the number of joints considered in the modelisation, a 1xn $\overrightarrow{q}$ vector is chosen to describe system possible configuration. The EE's pose vector $\overrightarrow{x}$ is defined as well (see fig. 2.9 for this case study).
- **Step 1:** parallel loops are opened (see fig. 2.11), operating three "cuts" in correspondance of points A, B and C. In this way, the moving platform results isolated from the legs.
- **Step 2:** as relations between A, B and C points are fixed and known (through Blender, see Appendix A), a unique solution can be found for the three identical kinematic chains thus obtained. Through an homogeneous matrix approach, positions vectors p of points A, B and C are expressed, each in its Local Reference Frame (LRF):

$$\overrightarrow{p_A^L} = f(\overrightarrow{q_A});$$
$$\overrightarrow{p_B^L} = f(\overrightarrow{q_B});$$
$$\overrightarrow{p_C^L} = f(\overrightarrow{q_C}).$$

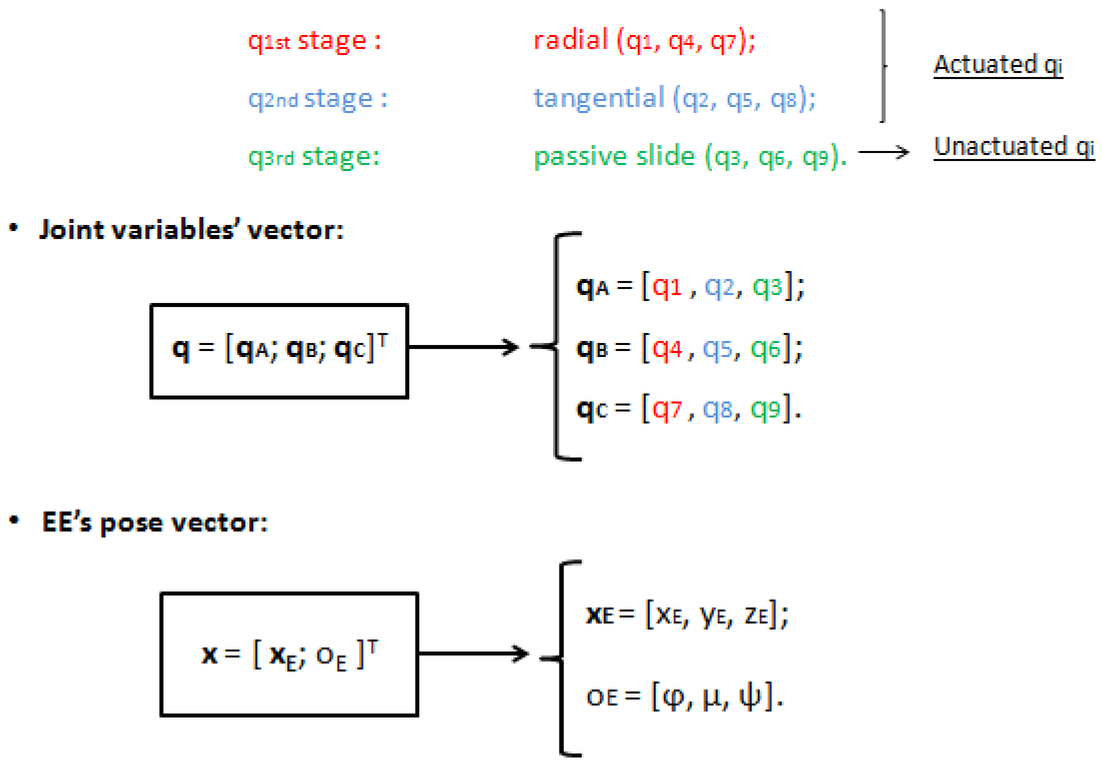


Figure 2.9: Definition of joint variables and of EE's pose in SmarPod's case study.

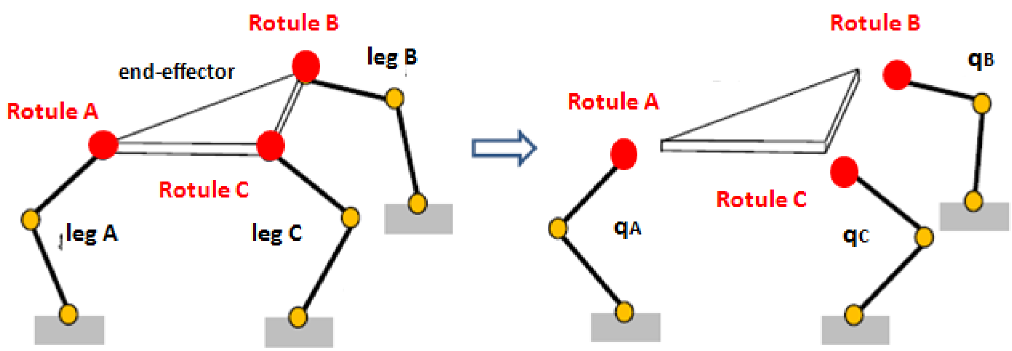


Figure 2.10: From (Bicchi et al., 2008): operating cuts on a 3-legged parallel robot, decoupling legs and moving platform.

- **Step 3:** obtaining vectors $\overrightarrow{p}$ in which coordinates of rotules are expressed in the Global RF (G) is now possible. In fact, fixed and known relations between E reference point and rotules exist (vectors $\overrightarrow{o_A}, \overrightarrow{o_B}$ and $\overrightarrow{o_C}$ to shift from Global to Local RF are known by geometry):

$$\overrightarrow{p_A^G} = \overrightarrow{p_A^L} + \overrightarrow{o_A};$$
$$\overrightarrow{p_B^G} = \overrightarrow{p_B^L} + \overrightarrow{o_B};$$
$$\overrightarrow{p_C^G} = \overrightarrow{p_C^L} + \overrightarrow{o_C}.$$

- **Step 4:** closure equations are imposed, restoring the initial closed loop through the expression of rotules' points as functions of the pose of the EE. In this case only the first three elements of the $\overrightarrow{x}$ vector are considered, as only positioning is concerned: they're collected in $\overrightarrow{x_E}$ vector.

$$\overrightarrow{p_A^G} = g(\overrightarrow{x_E});$$
$$\overrightarrow{p_B^G} = k(\overrightarrow{x_E});$$
$$\overrightarrow{p_C^G} = w(\overrightarrow{x_E}).$$

- **Step 5:** three systems of equations are thus obtained and implemented in Matlab in their more compact matrix form: $Ax = b$, in which A is an mxn matrix, x is a column vector with n unknows, while b is a column vector of n known terms. Matrix A results to be invertible: therefore it's possible to obtain joint variables vectors $\overrightarrow{q_A}, \overrightarrow{q_B}, \overrightarrow{q_C}$, as Matlab is capable of solving systems of equations put in the previous compact form.

2.4.2 Orientation

In the first istance, to deal with the second part of the IGM, a representation of orientation has to be selected, which aids in order to understand in which order rotations have to be applied on the moving platform.
In robotics, various representations of orientation are employed, which can be minimal or not. The most used are the following (Bicchi et al., 2007):

- **rotation matrix** is a redundant (non-minimal) representation of orientation, as its 9 elements are not independent betwen them. Representing rotations thorugh a 3x3 matrix can lead to futher drawbacks. In fact, representing rotations employing coordinates' expressions of axes' versors is not convenient; in addition, it lacks in robustness. Therefore, it is important to have other representations of orientation, especially those employing only 3 parameters, called minimal;

- **Euler's angles** is a typical minimal representation of orientation, as only three angles of rotation are involved, each relative to a precise axis of rotation. Various configurations exist, depending on how the rotation's sequence is chosen: the concept is to start with a fixed reference frame and then performing three rotations about current axes;

- **axis-angle** representation is not minimal, because it employs 4 parameters: axis' versor **r** and a rotation angle. Moreover, they are not independent between them as ${r_x}^2 + {r_y}^2 + {r_z}^2 = 1$. It is usually chosen in trajectory-planning to orient a manipulator's EE;

- **quaternions** representation is comparable to axis-angle one: a quaternion can be considered a generalization of complex numbers and can be defined as: $\overrightarrow{q} = a + b\overrightarrow{i} + c\overrightarrow{j} + d\overrightarrow{k}$, in which a in a scalar component, while b, c and d are the so-called Euler's parameters

According to (Merlet, 2008), parallel robots' orientation is typically described by Euler's angles or by Roll-Pitch-Yaw (RPY) angles. An Euler XYZ representation of orientation has been selected in this case study, to stay coherent with Blender's implemented one. Rotation is an **isometric transformation**: it preserves angles and distances. The idea is to take advantage of this feature and, starting from rotules' initial known positions (derived from Blender's model), to obtain final rotules' positions, after having applied a given rotation to the moving platform. The selected pivot point will be, as usual, E, platform's central point. The main interest, in this sense, is to understand all possible orientations that can be reached while E is in a fixed location: this means to work in the **orientation workspace**.
The workflow to solve IGM in case of rotations applied on the hexapod platform is explained below:

- **Step 0:** a representation of orientation has been chosen;

- **Step 1:** according to Euler XYZ convention, three angles of rotation are imposed to the EE. First a ϕ angle about x-axis, then a θ angle about y and finally a ψ angle about z. Resulting Euler's matrix, expressed in current axis, is:

$$R_x(\varphi)R_y(\theta)R_z(\psi) = \begin{bmatrix} c\theta c\psi & -c\theta s\psi & s\theta \\ cvarphis\psi + c\psi s\varphi s\theta & c\varphi c\psi - s\varphi s\theta s\psi & -c\theta s\varphi \\ s\varphi s\psi - c\varphi cpsis\theta & c\psi s\varphi + c\varphi s\theta s\psi & c\varphi c\theta \end{bmatrix}$$

- **Step 2:** Euler's matrix found above is applied to rotules initial positions in order to obtain final ones after the applied rotation.

$$\overrightarrow{p_{Af}} = R_x(\varphi)R_y(\theta)R_z(\psi) \times \overrightarrow{p_{Ai}}$$

$$\overrightarrow{p_{Bf}} = R_x(\varphi)R_y(\theta)R_z(\psi) \times \overrightarrow{p_{Bi}}$$

$$\overrightarrow{p_{Cf}} = R_x(\varphi)R_y(\theta)R_z(\psi) \times \overrightarrow{p_{Ci}}$$

- **Step 3:** Once rotules' final position have been obtained, returning to step 3 of the positioning part is now possible in order to solve IGM and acquire joint variables' values. A Matlab implementation of the previous relation has been performed too, in order to calculate $\overrightarrow{q_A}$, $\overrightarrow{q_B}$ and $\overrightarrow{q_C}$.

Chapter 3

THE ROLE OF VIRTUAL REALITY IN MODELING A PARALLEL MICROMANIPULATOR

3.1 Blender software overview

Blender (*Blender foundation*) is a free and open-source software mainly employed for 3D computer graphics, animations, visual art, video games and interactive 3D applications. It employs a **virtual scene**, whose fundamental elements are *the Object, the Camera and the Light Source* (see fig. 3.1). The basic Blender's working principle consists in calculating the virtual image seen by *the Camera* through adequate mathematical and physical models, in order to represent the interaction between *the Light* and *the Object* (Haliyo, 2014).

Blender's main features and capabilities are the following:

- **photorealistic rendering**, realized thanks to the creation of bodies and meshes with realistic materials;
- **fast modeling**, whose computational time needs to find a trade-off between a realistic outcome and the need to provide a real-time performance;
- **animation toolset**, as the software was initially developed for this purpose;
- **simulation tools** for soft or rigid body dynamics: for example, fluids, collision detection and ocean generator with waves can be performed;
- Blender possesses an integrated Game Engine, (**Blender Game Engine BGE**, writted in C++), a tool for real-time projects which allows to fully conceive and

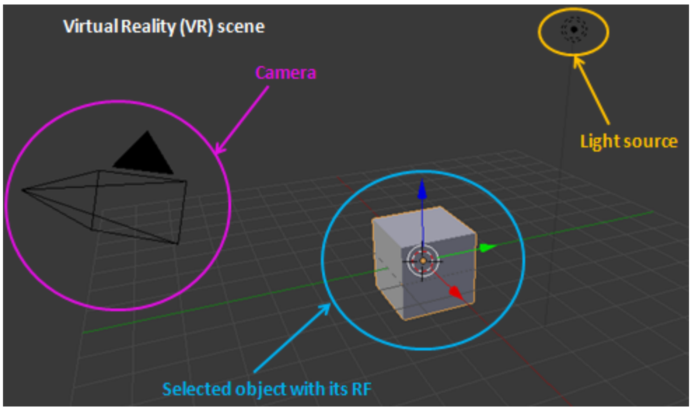


Figure 3.1: Blender's basic components of the virtual scene: *the Object*, shown with its own Reference Frame, the *Light Source* and *the Camera*

realize a 3D game in Blender or to create stand-alone interactive applications. BGE can be employed in each step of game design, from the prototyping to the final game simulation. The most important difference between conventional Blender and BGE lies in the *rendering process*: in BGE, in fact, scenes are continously rendered in real-time, allowing user's interaction during the process itself. In conventional Blender, animations are built offline and the user can't actually modify them. To enable interactions with the 3D Object in the virtual scene, BGE requires to arrange the so-called *Logic Bricks*, fundamental components of the *Logic Editor* (see next section for details), before lauching the Game.

This case study represents an example of the **essential role of Virtual Reality** when dealing with micro-scale systems. For thesis' purposes, Blender most interesting feature is the possibility to graphically visualize kinematic chains, called *armatures*, fully provided with their links (called *bones*) and joints; in addition, imposing positions, velocities and acceleration to a rigid body within the whole architecture is possible, thanks to the implementation of **Python scripts**. Blender integrates them in the *Game Engine*: scripts are executed at each step of the simulation.
Blender's animations allow to simulate SmarPod's motion patterns, before performing them on the real robot, enhancing micromanipulation's **performances** and system's **safety**. Trajectories and potential outcomes are analyzed as well, avoiding the risk of damaging both the delicate architecture of the micropositioner and the internal set-up of the SEM. In fact, SmarPod is designed for vacuum-applications: inside a SEM's chamber it's hard to have a clear view of the whole system, as dimensions of the robot are generally bigger than those of the sample. Thus, the robot can't be visualized entirely through the microscope.
Therefore, a complete exploration of positioning and manipulation strategies is performed.

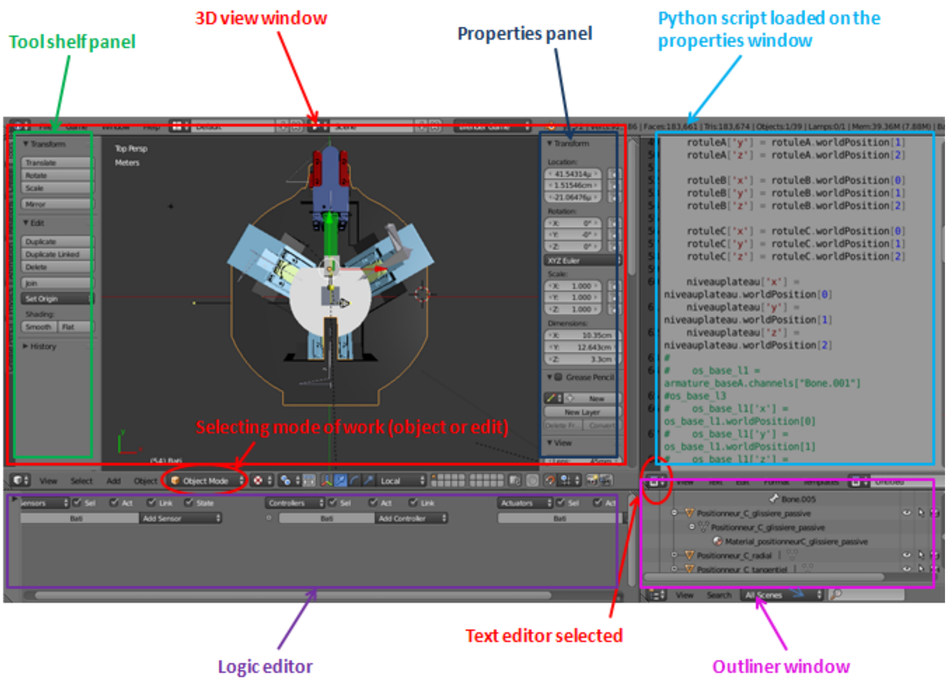


Figure 3.2: Blender user interface with SmarPod model and its relative Python code loaded. Blender's windows employed in this case study are shown as well

Interesting features concerning the robot's inverse kinematics are extracted as well, along with a clear representation of robot's architecture.

3.1.1 User interface

Blender user interface (see fig. 3.2) is window-based and customizable: all sections can be resized and possibly removed.

Two main modes of operation exist in Blender environment, named **object mode** and **edit mode**. With the first one, which is usually enabled by default, the selected *Object* is treated as a whole rigid body. It can be selected to be moved around in the virtual scene, to be scaled or rotated or to be grouped together with other *Objects*, etc. With the second mode, on the contrary, it's possible to change the content of an *Object*, in terms of its materials, vertices, volume, control points for curves/surfaces, etc.

If Blender is employed in the *default* mode, visualized windows are:

- **3D View** window, which is the heart of Blender's software and it is used to interact with the 3D scene for various purposes, such as modeling, animation, etc. This window includes two important panels: the **tool panel**, which allows to move, rotate and scale the selected *Objects* and the **properties panel**, which includes the description of transformations applied to *Objects*, current scales and other interesting features about *Object*'s display;

- **Properties** window, on which, for example, Python scripts can be loaded (through the *Text editor* button) or properties, such as objects' constraints, materials, physics and scene's details, can be visualized. Disposing of Python scripts results a very convenient feature, as they convey the communication with the external world: for example, through a Python script it would be possible to succeed in a real-time **control** of a robot 3D-model.

When selecting the *Game Logic* mode instead of the *default* one, furter useful windows are available on the interface:

- the **Outliner** window is very useful when, as in the SmarPod's case, lots of objects are included in the scene, as they're all listed in this area: it's possible to hide or to show items, to select or to deselect them in the virtual scene, etc. In the Outliner, each line represents a data block, expandible through the plus-sign at the begininning of the row (the Outliner window for the SmarPod model is shown in fig. 3.3);

- the **Logic Editor** is fundamental in order to manage the *Game Logic*: it provides interactions with the virtual scene and its functionalities can be enhanced thanks to Python scripts' integration. This high level, event-driven window is made up of *Logic Bricks*, arranged in three columns corresponding to Sensors, Controllers and Actuators. These three areas result linked between each other to show the logical flow between sensors-controllers and controllers-actuators. In fact, a sensor can't directly be linked to an actuator withouth passing through the relative controller block. In the Sensors' area they are listed all the sensors owned by the selected object; the same thing is valid also for Controllers and Actuators. Logic Editor also includes *Game Property* area: in Blender, game properties are like variables in other programming languages. This section will be very useful in order to extract SmarPod's positions of the rotules, as game properties are employed to access data associated with an *Object*.

3.2 Importing SmarPod's model

Once essential features of Blender software have been clarified, it's possible to load on it an available SmarPod model, in order to explore possible Blender-based strategies in the purpose of solving the IGM and of discovering new details about this case study.
Figures 3.3 and 3.4 illustrate how both the *Outliner* and the *Logic Editor* (relative to

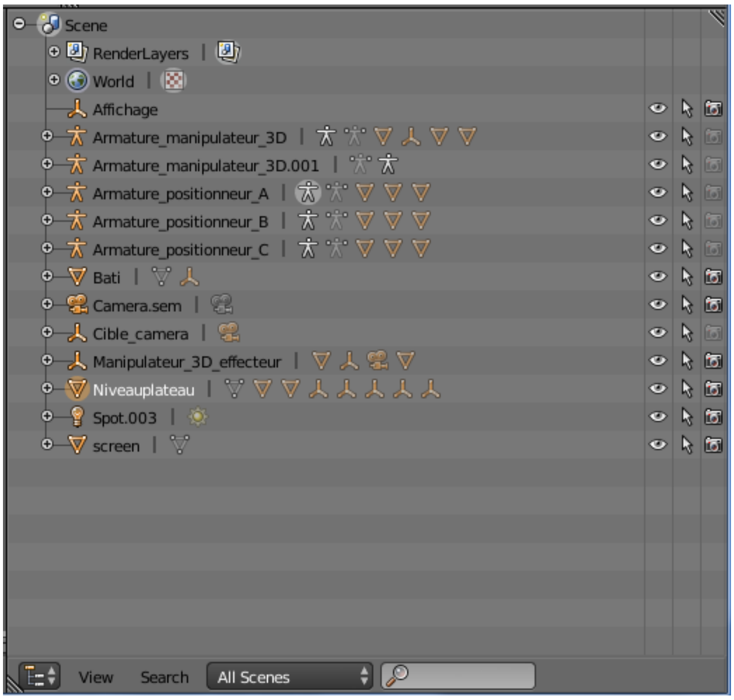


Figure 3.3: The expanded Outliner window when SmarPod's model is loaded (the selected block *niveauplateau* represents the moving platform)

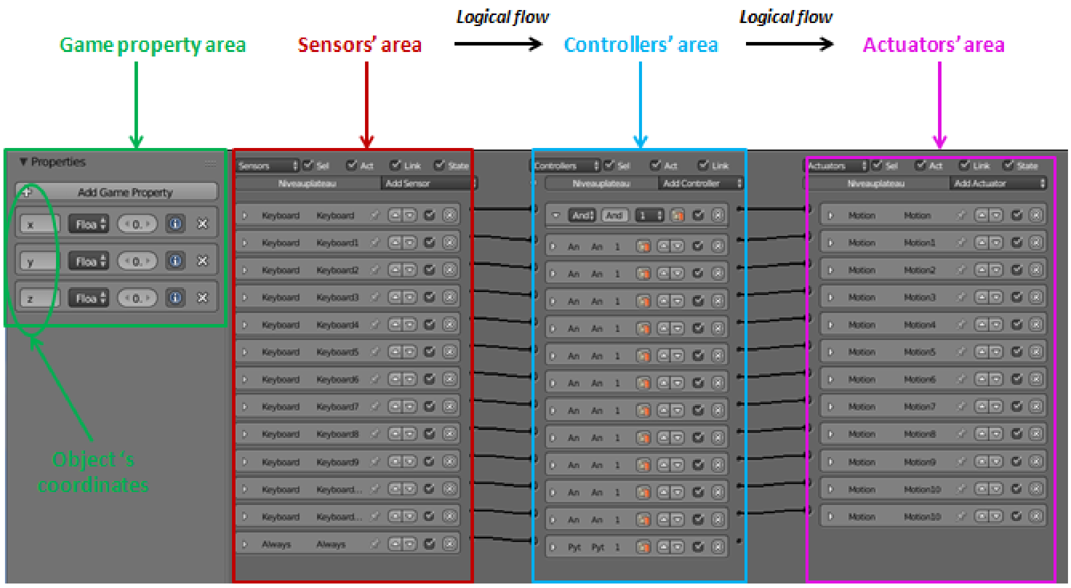


Figure 3.4: Blender's Logic Editor relative to moving platform when SmarPod's model is loaded

the moving platform) appear when the model is opened.
The **Outliner** lists all elements included in the model: this organization allows the user to proceed with operations in an intuitive and fast way, easily selecting robot's structures of interest. For this study's purposes, the main components employed in the showed model will be all the *armatures positionneurs*, in other terms A, B and C kinematic chains, along with the *niveauplateau*, which corresponds to the moving platform.
In fig. 3.4 the **Game Property** area is illustrated, which will be essential to deal with the IGM; the logical flow existing among sensors', controllers' and actuators' areas is underlined as well.
For these thesis purposes, Blender's tools described above will be useful to write a new Python code, in order to **extract joint variables** from the implemented SmarPod's model: this will be fundamental in Chapter 4 to allow a first A-IGM's validation. To obtain the set of joint variables, however, a preliminar step is performed: initial positions of points A, B and C (in other terms, positions of **rotules**) referred to the Global Reference Frame (GRF) will be recovered thanks to the development of another Python script. These coordinates will provide important information about the geometrical relations existing between EE's points (see Appendix A).

3.2.1 Recovering positions of rotules

Coordinates of passive spherical joints (also called *rotules*) A, B and C, expressed in the Global Reference Frame (GRF), are needed in the Analytical IGM to write **closure equations** in the *positioning* subsection (see Chapter 2). Moreover, they're essential in order to recover geometrical distances and, consequently, to derive useful relations between points of the EE. They will be employed in the *orientation* subsection of the A-IGM as well: Euler's matrix is applied to these initial coordinates in order to calculate the final ones after rotations (expressed in the GRF, see Chapter 2).
For these aims, a new adequate strategy is implemented on Blender in this thesis in order to recover them; the employed approach consists in the following steps:

1. The first step is to take advantage of the *Game Property* area on Blender interface, by **adding game properties** to passive spherical joints. In Blender environment, game properties are analogous to features like point's coordinates: so, for each of the three rotules (from the *Outliner: Plateau -¿ Niveau Plateau -¿ access to Rotule A, Rotule B, Rotule C*), three game properties will be added, corresponing to x, y and z coordinates, relative to the GRF.

2. Writing a **Pyhton script** (the complete and commented Pyhton code is available in Appendix B) is now necessary to calculate and print coordinates of rotules. In Blender, recovering features such as position and orientation of an *Object* is generally simple, as they are represented in a specific Pyhton class.
   In order to include a script into a Blender's simulation, the preliminar step consists

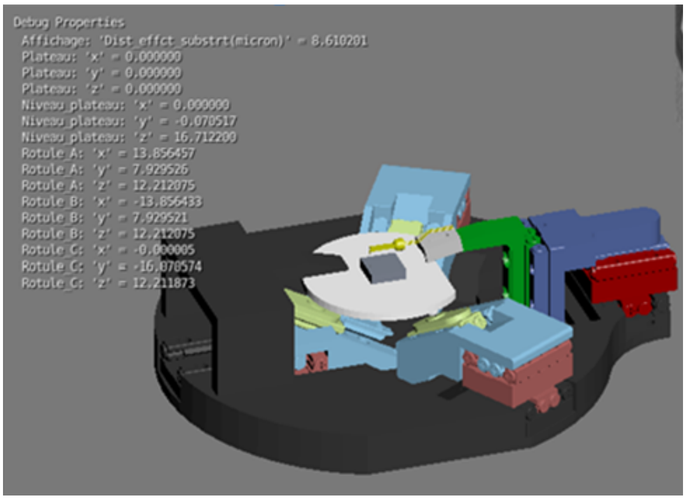


Figure 3.5: Initial positions of rotules, recovered and displayed on Blender

in adding a Pyhton controller, linking it to any sensor block.
Then, at the beginning of each script, access to the whole scene must be provided:

$$scene = logic.getCurrentScene()$$

Then, *Objects* (such as the moving platform) and *Sub-Objects* (such as armatures' bones) of interest have to be accessed in the virtual scene. For example, if an object named *"Obj"* and listed in the Outliner as *"Object name"* has to be get from the scene, the employed instruction is:

$$Obj = scene.objects["Object\ name"]$$

3. Actual coordinates of the *Object* in the VR are now available through the following instructions:

$$Obj['x'] = Obj.worldPosition[0];$$
$$Obj['y'] = Obj.worldPosition[1];$$
$$Obj['z'] = Obj.worldPosition[2];$$

4. To display positions of rotules, the Game Engine has to be selected (*Game -¿ Start Game Engine*). An animation is launched and results are printed in the 3D view, as shown in fig. 3.5

3.3 Simulation-Aided IGM

Taking advantage of Blender *Game Engine*, it's possible to obtain a more efficient and intuitive solution of the SmarPod geometrical model, without resorting to the classical method employed in Chapter 2 to derive A-IGM. In this case, no complex calculations or homogeneous matrix will be needed, as a direct expression of **joint variables** is found. Once again, assuming the linearity of the system, the **superposition principle** is applied, allowing to treat separately translations and rotations of the moving stage.

3.3.1 Position

The main idea is to launch the **Game Engine**, starting in this way an animation. Then, desired single-axis translations are imposed to the moving platform; motions of the positioners on the three stages are observed, in order to find simple expressions representing each joint variable as function of the $x-y-z$ displacement imposed on the EE.
In this way, single contributions of translations along x, y or $z-axis$ for each SmarPod's stage are found; then, they're summed up into a unique expression of each joint variable. This approach results another application of the **superposition principle**: for example, a translation along $x-axis$ of a d_x quantity is considered. Motions performed by all positioners in order to achieve the imposed translation are studied and three simple trigonometrical relations between $q_{1st}, q_{2nd}, q_{3rd}$ values and d_x displacement (referred to GRF) are found. Then, the same method is repeated for single y-displacements d_y and z-displacements d_z, always referred to GRF. Displacements applied d_x, d_y, d_z are collected in a $\overrightarrow{d}$ vector.
At the end, the following joint variables' vectors are found (for details, see Appendix A):

$$\overrightarrow{q_A} = [d_x cos(\delta) + d_y sin(\delta) + d_z\sqrt{3}; d_x sin(\delta) - d_y cos(\delta); d_z \frac{1}{sin(\alpha)}]$$

$$\overrightarrow{q_B} = [-d_x cos(\delta) + d_y sin(\delta) + d_z\sqrt{3}; d_x sin(\delta) + d_y cos(\delta); d_z \frac{1}{sin(\alpha)}]$$

$$\overrightarrow{q_C} = [-d_y + d_z\sqrt{3}; d_x; d_z \frac{1}{sin(\alpha)}]$$

These relations are implemented in Matlab to recover $\overrightarrow{q_A}, \overrightarrow{q_B}, \overrightarrow{q_C}$.
From this analysis, interesting features about SmarPod's motion patterns are extracted:

- passive guides are not involved into translations on the xy plan: in other terms, $q_{3rd} = 0$ for this kind of motion and only the first and the second stages are necessary to perform the motion. On the contrary, passive slides participate when the EE is translated along the $z-axis$. In the latter case, the second stage does not intervene ($q_{2nd} = 0$);

	Chain A	Chain B	Chain C
d_x	d_{A1x}	d_{B1x}	d_{C1x}
d_y	d_{A1y}	d_{B1y}	d_{C1y}
d_z	d_{A1z}	d_{B1z}	d_{C1z}

Table 3.1: In the SA-IGM orientation section, it's necessary to substitute components of general displacement used for translation with rotule-specific displacements

- looking at the $\overrightarrow{q_C}$ vector, so considering chain C, it appears that, for EE's translations on xy plan, the first radial stage performs motions along $y-axis$: being radial, its positioner is aligned to the $y-axis$ of the GRF. On the other hand, the second stage performs motions on the $x-axis$: being tangential, the corresponding positioner is aligned to the $x-axis$

3.3.2 Orientation

Starting from the same choice about representation of orientatio made in Chapter 2, the final positions of passive spherical joints (for example, for rotule A: $\overrightarrow{p_{Af}} = (p_{Afx}, p_{Afy}, p_{Afz})$) in the GRF are found, thanks to the application of $XYZ-Euler$ matrix to initial positions of rotules ($\overrightarrow{p_{Ai}} = (p_{Aix}, p_{Aiy}, p_{Aiz})$).
The employed approach to solve SmarPod's IGM in case of multiple rotations is to **deal with rotations such as translations**.
Once vectors $\overrightarrow{p_{Af}}$. $\overrightarrow{p_{Bf}}$, $\overrightarrow{p_{Cf}}$ are recovered, it is possible to calculate the net displacement vectors of each rotule (named $\overrightarrow{d_{A1}}, \overrightarrow{d_{B1}}, \overrightarrow{d_{C1}}$), as differences between final and initial positions of rotules, after the first applied rotation (that is why the "1" subscript has been employed). Considering, for example, rotule A, its resulting displacement vector will be composed as follows:

$$\overrightarrow{d_{A1}} = [p_{Afx} - p_{Aix}; p_{Afy} - p_{Aiy}; p_{Afz} - p_{Aiz}]$$

Expressions of $\overrightarrow{q_A}, \overrightarrow{q_B}, \overrightarrow{q_C}$ joint variables found in the translation section are then used, with substitutions listed in table 3.1, in order to complete the solving of SA-IGM (Simulation-Aided IGM). In fact, while in translation the applied displacement vector $\overrightarrow{d}$ is the same for all chains, when rotations are applied about point E (see Appendix A) they result in different net displacements for each chain in the GRF (vectors $\overrightarrow{d_{A1}}, \overrightarrow{d_{B1}}$ and $\overrightarrow{d_{C1}}$, after the first rotation).
Finally, obtained relations are implemented on Matlab in order to extract $\overrightarrow{q_A}, \overrightarrow{q_B}, \overrightarrow{q_C}$ vectors. If multiple rotations are applied, it's sufficient to repeat the steps shown above.

Chapter 4

MODELS' VALIDATION AND RESULTS

In the present chapter, two different validations of the Analytical Inverse Geometrical Model (A-IGM) and of the Simulation-Aided Inverse Geometrical Model (SA-IGM) will be provided. In the first section, Blender software, whose properties have been fully investigated in Chapter 3, will be employed to conceive an adequate strategy to recover the values of joint variables (the $\overrightarrow{q}$ vector) on SmarPod's virtual model. The *virtual* joint variables will be then compared with those obtained through A-IGM and SA-IGM. In this way, the most Blender-coherent model will be experimentally validated on the real robot using a microrobotic cell. The workflow is depicted in fig. 4.1

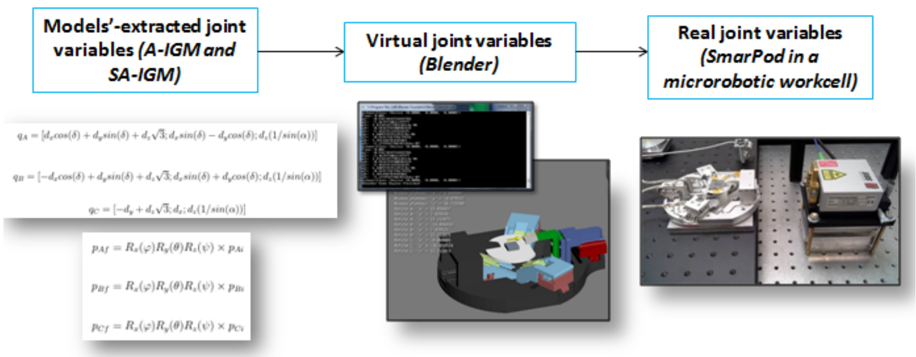


Figure 4.1: Workflow followed for models' validation

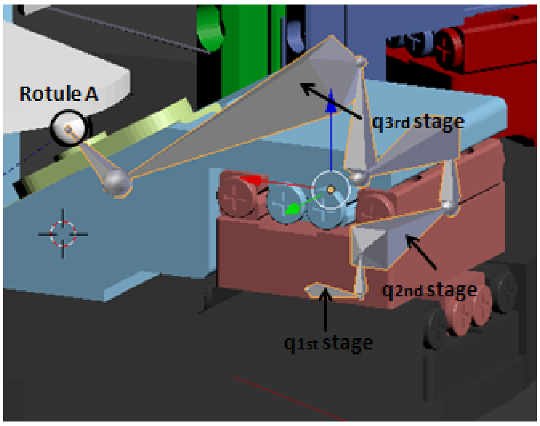


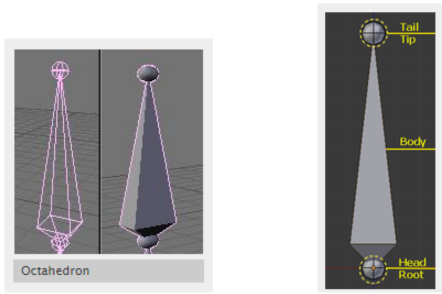
Figure 4.2: Armature A with its joint variables', underlined on armature's bones

4.1 Blender validation

In this section, a method to recover joint variables on Blender will be developed, along with the necessary Python instructions to obtain them (the complete and commented Python script can be found in Appendix B). Blender software, in this case study, is employed in order to achieve a **preliminar validation** of A-IGM and SA-IGM. The most accurate model between the two extracted will be selected thanks to this preliminar validation. Then, its relative sets of joint variables will be compared with those extracted from the real robot.

4.1.1 Extracting joint variables

The aim is to find configurations of joint variables corresponding to given translations/rotations applied on the EE. This section deals with **solving the IGM in Blender environment**, which will be used as a first validation of both the A-IGM and the SA-IGM.
The IGM problem has already been defined in Chapter 2 and, assuming that geometrical properties of SmarPod's model implemented on Blender are very precise, as it is based on a detailed available CAD model, recovering the $\overrightarrow{q}$ vector through the software will represent a first **validation method** for models found above.
Observing the architecture of one of the three armatures (illustrated in fig. 4.2) as depicted on Blender, it will be clear that an adequate strategy has to be developed in order to recover joint variables.
In fact, **armatures** basically mimic articulated skeletons, both in structure and behaviour. They're made up of **bones**, which consist in a start point, named *head*, the bone's body itself and the end point named *tail*. Bones can be represented in various shapes (the adopted bone's shape representation in this case study is reported in fig. 4.3).



(a) One of the possible bone's visualization

(b) Bone's elements

Figure 4.3: Bones' visualization on Blender

In this scenario, the idea for recovering joint variables is to represent them as **distances between the head and tail of a selected bone**. Nine bones will be selected, one for each q_i: in particular, the ones lying in the same direction of the desired q_i will be taken into account.
When a motion is imposed to the EE, actuators move following a precise pattern in order to achieve the required EE's configuration and poses of a single bone's head and tail change in Blender, depending on the experimented displacement.
Consequently, joint variables are computed as differences between poses of the head and the tail of bones, each belonging to the bone which lies in the direction of motion of the required joint variable. This value corresponds to the net displacement experimented by each positioner after having applied an input on the EE: in other terms, joint variables are obtained.
If required bone's name (to be read in the Outliner) is $bone_i$ and it corresponds to joint variable q_i, the needed instruction is:

$$q_i = (bone_i.posehead - bone_i.posetail).length$$

This command will be repeated 9 times, one for each joint variable.
However, to obtain q_i values different from zero, a movement has to be applied on the moving platform, in terms of a translation or a rotation along or around the xyz GRF axes.
Translations are imposed by keyboard after having started the animation (shortcuts shown in table 4.1). However, through the Blender sensor area, shortcuts can be arbitrarily changed.
Rotations, on the other hand, are obtained thanks to a proper set of instructions and not by keyboard (even if imposing them by keyboard should be possible, as it has

	positive displacement	negative displacement
along x-axis	left arrow	righ arrow
along y-axis	up arrow	down arrow
along z-axis	T	Y

Table 4.1: How to impose EE's translation from keyboard on Blender

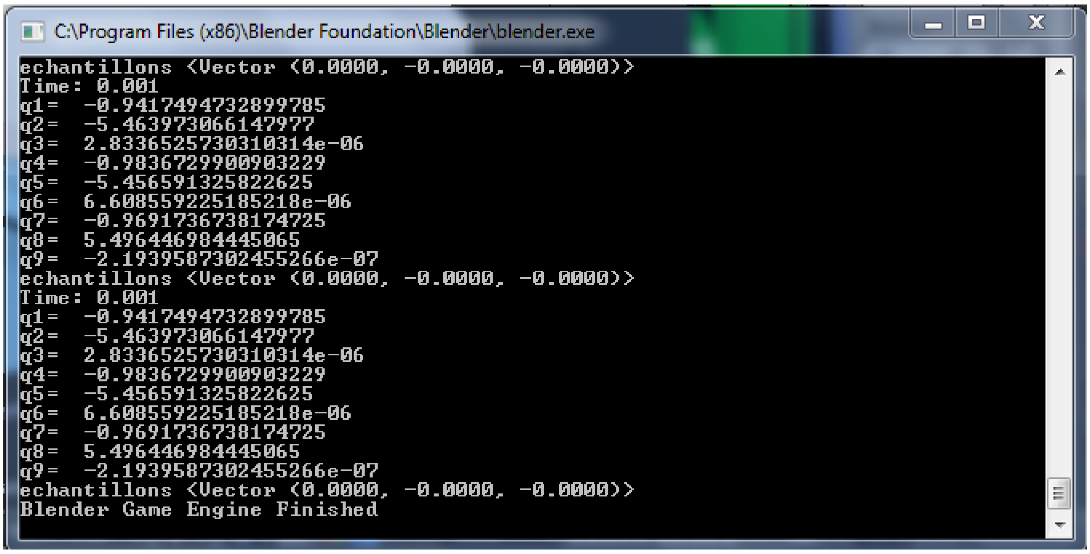


Figure 4.4: Joint variables' values printed for a 20 deg rotation applied about the z-axis

been done with the translation). If, for example, a rotation about $x - axis$ of deg degree is desired to be applied on the *plateau* (the EE), required commands are the following:

$$\textit{anglex = deg*math.pi/180;}$$
$$\textit{plateau.applyRotation([anglex, angley, anglez])}$$

To visualize the resulting q_i, resulting q_i have to be printed using the Python instruction:

$$print('q_1 =', q_1 - logic.q_1),$$

where $logic.q_1$ has been computed in the script in order to avoid offset's accumulation. It occurs when repeating the animation multiple times: in fact, Pyhton script is compiled within each simulation. Also in this case, the istruction is repeated for each joint variable. Finally, $Window- > ToggleSystemConsole$ has to be selected, then the $GameEngine$ needs to be started: joint variables' values are displayed, as shown in fig. 4.4.

4.1.2 Preliminar results and discussion

This preliminar Blender validation is useful in order to understand which model between the A-IGM and the SA-IGM is more accurate, making a comparison of their joint variables' values with Blender ones. Results are analyzed separately for rotations and translations. Full results for various examples of motions applied to the EE are provided in Appendix C.
After this step, the experimental validation on the most Blender-coherent model will be performed. In fact, this idea is based on the assumption that SmarPod model's implemented on Blender is extremely precise, being derived from a CAD-model of the robot itself. Consequently, also the implemented inverse kinematics should precisely mimic the IGM with which the robot itself has been developed. When applying rotations on the moving platform, one must take into account SmarPod's full-travel ranges: with the instruction employed on Blender, any angle of rotation can be set through the Python script. When exceeding rotation limits shown in tab. 4.2, Blender provides the opportunity to visualize what would happen to the micropositioner, as these full-travel ranges haven't been implemented. A detachment of the moving platform (shown in fig. 4.5) occurs and resulting joint variables are no more coherent with any of the two models proposed. If such exceeding rotations had been applied on the real SmarPod, a blocking mechanism would have prevented such a damage to occur.

X: 10 [mm]	θx: 17 °
Y: 10 [mm]	θy: 20 °
Z: 5 [mm]	θz: 35 °

Table 4.2: SmarPod's 115.25 full-travel ranges

However, for both translations and rotations only the SA-IGM provides Blender-coherent results for what concerns $\overrightarrow{q}$ vector. Being Blender-inspired, displacements' in SA-IGM correspond almost exactly, both for **single and multiple rotations**, to software outcomes. Moreover, results' coincidence valid also for multiple rotations proves that the selected representation of orientation *(Euler-XYZ)* is the correct one.
For what concerns A-IGM, the scenario appears to be more complicated: only correct rotule's final positions are obtained, while q_i values are not coherent with Blender. Their values exceed positioner's stroke. However, joint variables' expression, in the Matlab code derived, directly depend on rotules' final positions, which on the contrary are Blender-coherent. This result could appear unusual: a possible explaination lies in the IGM-problem definition, having treated the system as a **redundant** one, as further joint variables (those are relative to the third stage) have been added to simplify the resolution. While Matlab solves linear systems of equations in a purely mathemathical way, without

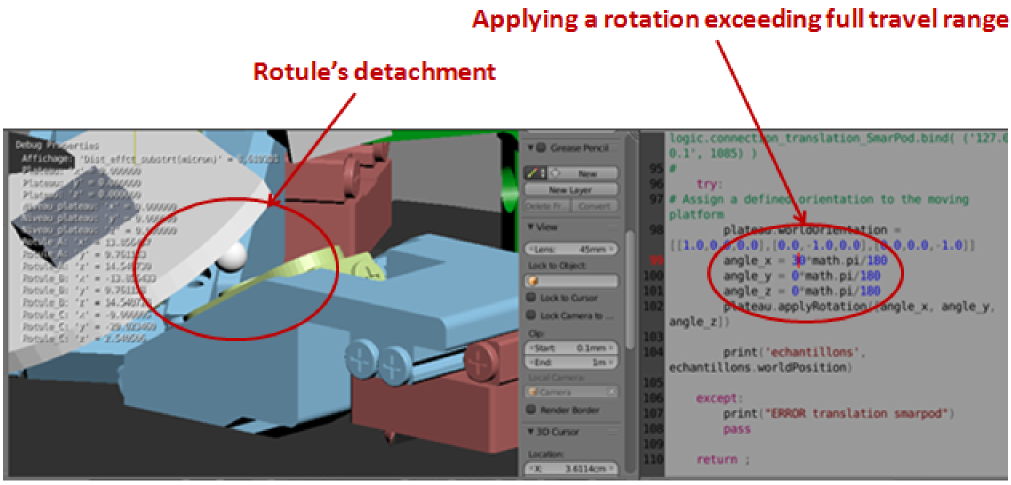


Figure 4.5: Rotule's detachment for an applied $x - axis$ rotation of 30 °

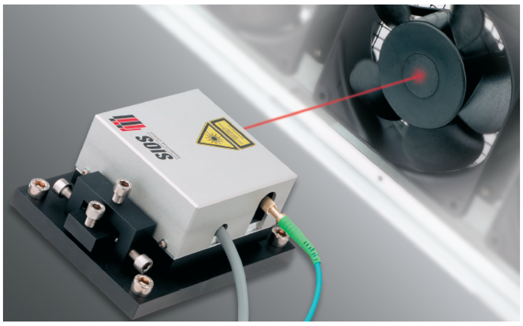
any link to the robot's implemented model, Blender appears to choose a more adequate unique solution in order to perform the same rotation. This could be a demonstration that many different combinations of joint variables values reach the same final orientation of the moving platform, when dealing with a redundant system.

4.2 Experimental validation

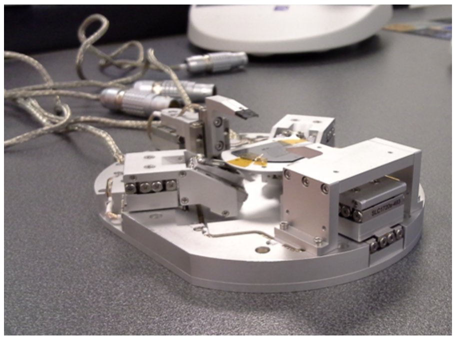
4.2.1 Experimental set-up

To validate SmarPod's SA-IGM on the real robot an experimental set-up is employed; experimental validation must be performed in a clean room, as microrobotic systems are very susceptible to environmental conditions and vibrations. In particular, a **microrobotic cell** (shown in fig. 4.7) is installed on an antivibrational table (MinusK®), which works taking advantage of the negative stiffness vibration isolators' principle. In addition, multiple shock absorbers are provided too: they are inserted between the SmarPod and the antivibrational table in order to absorb possible vibrations which could affect measurements. Single components of the experimental set-up are shown in fig. 4.6.

The microrobotic cell is made up of of the robot itself (see Chapter 2 for details) and of a laser interferometric vibrometer (SIOS Meβ technik GmbH SP-S 120). This particular interferometer is ideal for accurate, noncontact determination of temporal changes in positions of objects or surfaces. Therefore, it has been selected to **experimentally recover joint variables**, intended as positioners' displacements after some motion to the EE has been applied. The system is able to detect motions occurring along the



(a) Laser interferometric vibrometer (SIOS Meβ technik GmbH SP-S 120)



(b) SmarPod micropositioner (SmarAct GmbH)

(c) Antivibrational table's (MinusK®) adjustament

Figure 4.6: Main components of the experimental set-up

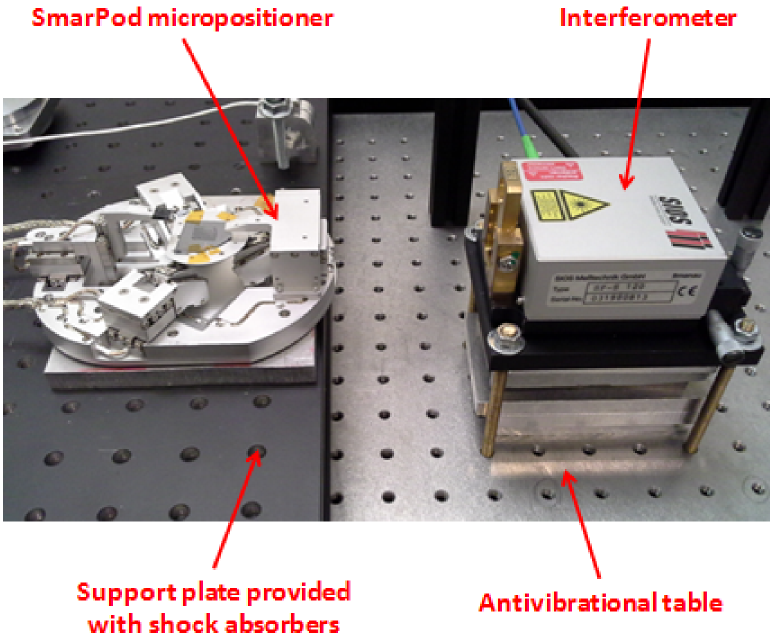


Figure 4.7: The complete experimental set-up, in the ISIR's clean room

optical axis of the interferomenter and to convert them into interference fringes, to be transmitted to electronics for signal processing. The latter, in this case study, will be performed on Matlab®.
The **idea** is to evaluate the displacement of **five** of the nine total positioners/passive slides considered in the model, taking advantage of the alignment of the interferometer's optical axis with the direction of motion of each actuator. Real values of (q_1, q_4, q_7) for the first stage and of (q_2, q_5) for the second one (see fig. 4.4) are therefore recovered. On the other hand, for what concerns the third stage, displacing and re-fixing the interferometer in such an inclined configuration result impossible. Neither q_8, on chain C second stage, can be accessed, because of a structure surrounding the whole chain: it prevents the interferometer's laser beam from reaching it at all.
Consequently, five different measurement configurations have to be found: in fact, for each recoverable joint variable, a precise alignment condition between the interferometer's laser beam and the positioner's line of action has to be obtained, in order to achieve precise interferometer's measurements. To find these experimental configurations, it's possible to resort to various strategies:

- Changing the orientation of SmarPod on the antivibrational table; thanks to a screwing grey plate it is possible, in a first instance, to obtain optical axis' alignment with the first stage's joint variables (q_1, q_4, q_7, see fig. 4.8);

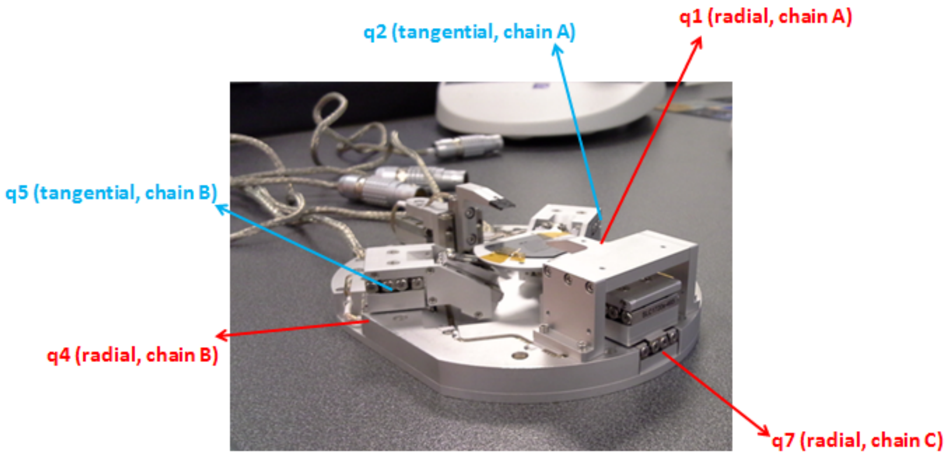


Figure 4.8: Experimentally recoverable SmarPod's joint variables (in red the first stage ones, in blue the second stage ones)

- the q_i of interest lie on stages at different heights: therefore, changing the SmarPod's orientation will not be sufficient to access to the second stage ones (q_2, q_5). The interferometer is provided with various levels to support it, enhancing its stability: its heigth can be changed by removing one or more of these stages and re-screwing the tool at the desired height. This last solution is employed, reaching the desirable alignment between second stage positioners and the laser beam.

Once experimental configurations of SmarPod have been decided, the same combination of traslations/rotations will be applied in each configuration: the protocol choice in this sense will be explained in the following section. Three measures will be taken for each positioner, in order to assure **repeatability** to the measurement itself.

4.2.2 Experimental protocol

Protocol selection The main aim here is to avoid a misalignment between the laser beam and actuators' lines of action, which could lead to significant measuring errors. Assuming that the chosen micropositioner's orientation and interferometer's heigth are the best to minimize a possible misalignment, the further step consists in selecting an adequate combination of translations and rotations to validate SA-IGM. To succeed in avoiding misalignments, first and second stage should not move at the same time; consequently, each stage could be studied independently from the other one.
Several possibilities have been evaluated through the virtual robot on Blender and the following combination has been chosen: **a set of two translations** ($-1mm$ each) and **three subtle rotations** ($1°$ each). Their order of magnitude (mm and $°$) has been

preferred to a smaller one in order to limit the noise influence on measurements.
The first translational part of the sequence is employed to show displacement in the **first stage actuators**, because second stage ones are not affected by this platform's motion (see tab. 4.3). For what concernes the second part of the sequence, it will provide only the **second stage actuators'** displacement, as the first stage ones will be only slightly affected by the rotation around z-axis (see tab. 1.4).
In this way, errors dues to misalignement between the interferometers' laser and actuators' direction of motion will be reduced in order to provide a valid experimental set-up to validate SA-IGM.

Translation	1st stage [mm]	2nd stage [mm]	3rd stage [mm]
Chain A	3.464	0	4
Chain B	3.464	0	4
Chain C	3.464	0	4

Table 4.3: *Table representing model's results in terms of final displacements of actuators for a simple translation on the z-axis = -2 mm*

Rotation	1st stage [mm]	2nd stage [mm]	3rd stage [mm]
Chain A	-0.0187	-0.836	0
Chain B	-0.025	-0.835	0
Chain C	-0.022	0.841	0

Table 4.4: *Table representing model's results in terms of final displacements of actuators for a simple rotation around the z-axis = 3 °*

Connecting the SmarPod To handle SmarPod's micropositioner, gloves are compulsory: possible dust on its surface could badly influence its mode of operation, even damaging the robot itself. Moreover, great care must be taken when displacing it: the moving platform and positioners are very delicate and they should not be directly touched. Fundamental steps to connect the SmarPod and to proceed with the experimental validation are illustrated below:

1. MCS controller's cables are connected as shown in fig. 4.9: an USB cable is needed to communicate with SmarPod's dedicated interface installed on a PC, along with the alimentation one and robot connectors.

2. On SmarPod's side, two among its three connectors must be connected to control the parallel robot, which are the A and the B: the C one is responsible of cartesian robot's motion and is not employed for this thesis' purposes. Then the micropositioner is screwed on a plate in the selected orientation configuration. To ensure a good alignment between the interferometer's optical axis and the positioner, their relative distance must be set at **7 cm**, which corresponds to **interferometer's working distance** found in its data sheet. The antivibrational table is then adjusted to ensure stability to the measurement.

3. SmarPod's UDP interface is now employed, as shown in fig. 4.10, to connect the SmarPod: in the first istance the robot is connected to the PC ("Connect"), then the EE is restored to its default position thanks to "Reference" button.

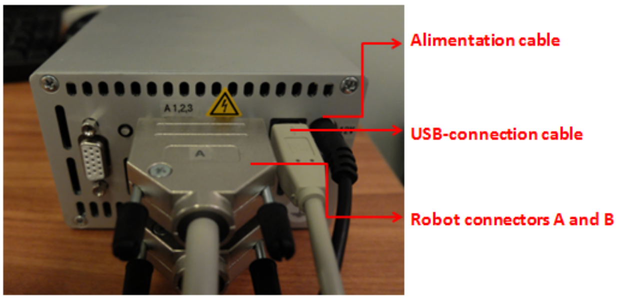
4. The interferometer is turned on. Its relative software is called INFAS: once the connection is set up, an important parameter called **signal monitor** must be evaluated as it provides an indication alignment goodness. For each configuration, it has to be maintained over **30 %** to ensure an adequate measurement: a preliminar calibration phase is thus performed before applying the experimental sequence to the EE.

5. Before recording data, the interferometer's modulator must be turned off, avoiding to obtain a sinusoidal outcome. *Fast Mode*, typical open-loop operation mode, is used: two parameters are set in its window, the **sampling frequency** and the **length of data set**. The first is set by default at 25 kHz. However, to perform the whole experimental sequence consisting in translation + rotation, a longer evaluation time is needed: consequently the frequency is set at 12.5 kHz. The length of data set consists in the number of points which are evaluated: it is employed the default one, 65536.

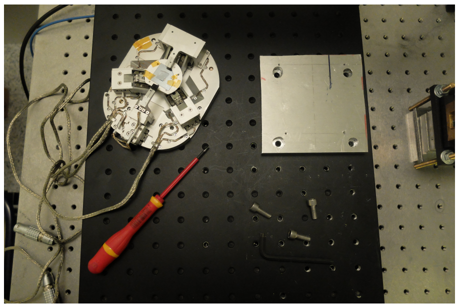
6. Before imposing the translation/rotation sequence on the EE through SmarPod's interface, *Fast Mode* is launched. Data is recorded in two formats: *.bsr* file contains useful info such as the selected sampling frequency, the length of data set and units of measurement, while the *.dat* one is loaded on Matlab, where extracted data's analysis is performed.

4.2.3 Results

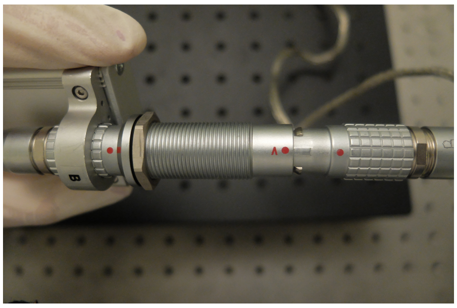
Experimental outcomes concering the two SmarPod's stages are presented separately: the **translation** part of the protocol is employed to study the first stage actuators (q_i of the second stage being always = 0), while the **rotation** sequence is useful to extract second stage's joint variables (q_i of the first stage are not equal to zero, but this rotation sequence is the one in which their displacement is minimized the most).



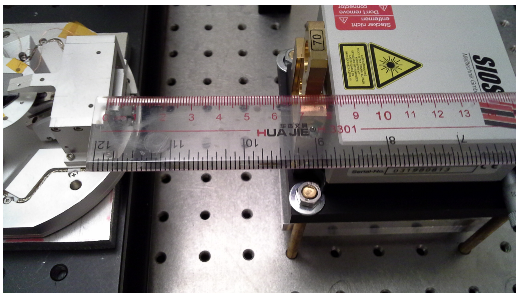
(a) SmarPod's MCS controller connection cables



(b) Mounting the SmarPod in the selected configuration on the screwable plate



(c) Connecting SmarPod's cables to MCS controller



(d) Setting the 7 cm working distance between SmarPod and interferometer

Figure 4.9: Preliminar steps to connect the SmarPod

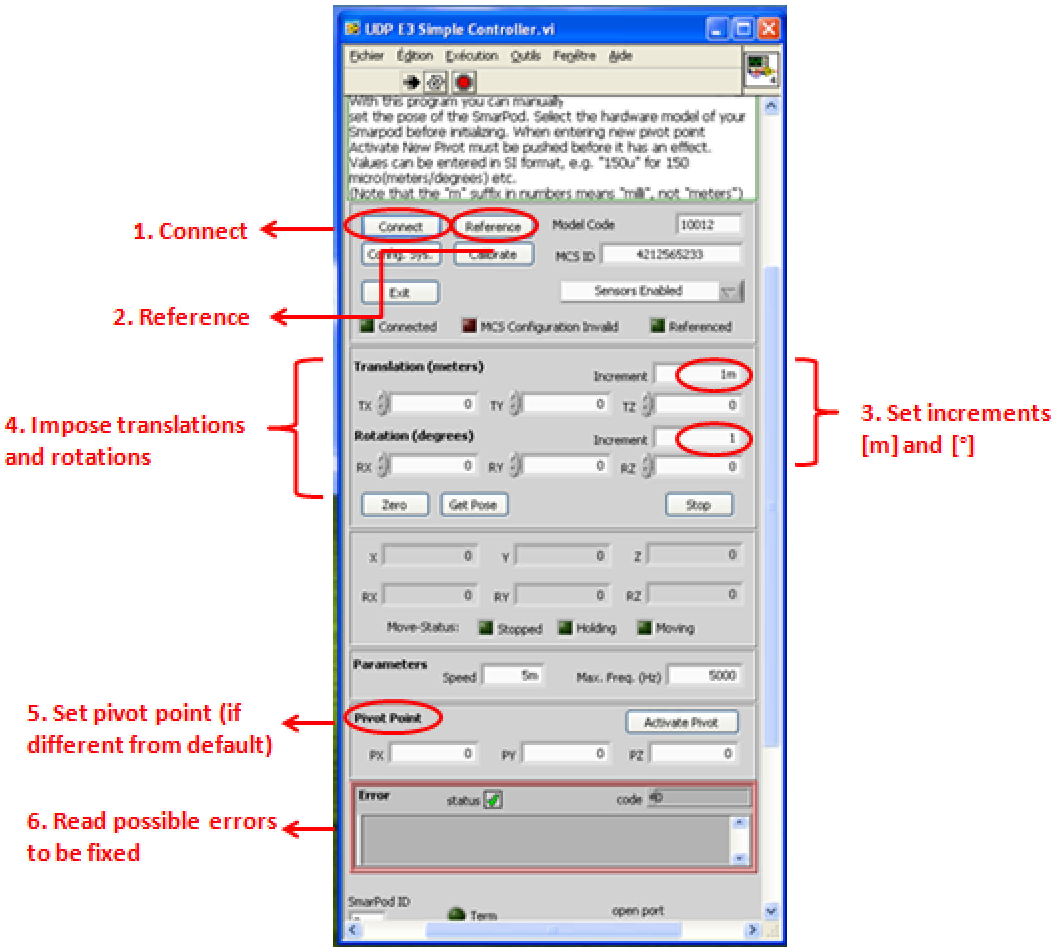


Figure 4.10: SmarPod's UDP PC interface and its relative workflow

Interferometer's outputs are recorded and then analyzed and plotted on Matlab in order to recover **joint variables**.
In the following sections, results concerning the two decoupled stages are presented:

First stage actuators They're only interested by the first part of the experimental protocol: the set of two z-axis translations, of $-1mm$ each, applied on the EE. Signals extracted from interferometer and plotted on Matlab are shown in fig. 4.11. Two steps in the first half of each plot are clearly remarquable, corresponding to the two subtle translations imposed to SmarPod moving platform.
As a preliminary step, the width of the medium step corresponding to a $-1mm$ translation is extracted from Matlab interferometer's data (see tab. 4.5), in order to compare it with the same parameter obtained from the Matlab SA-IGM implementation.

Medium step	SA-IGM [mm]	SmarPod [mm]
Chain A	1.732	1.6
Chain B	1.732	1.732
Chain C	1.732	1.723

Table 4.5: *Medium single step comparison, for an applied translation of -1 mm along z-axis, between SA-IGM and real SmarPod*

Total displacements for each chain (in other terms, **joint variables** q_1 for leg A, to q_4 for leg B and to q_7 for leg C) are then computed as differences between the recorded value before and after having applied the sequence. For each chain, 3 values are therefore obtained (one for measurements) and mediated. Final results are illustrated in tab. 4.6 and compared to SA-IGM analogous ones.

Joint variables	SA-IGM [mm]	SmarPod [mm]
q_1	3.464	3.2012
q_4	3.464	3.44
q_7	3.464	3.44

Table 4.6: *Joint variables comparison, for an applied total translation of -2 mm along z-axis, between SA-IGM and real SmarPod*

Second stage actuators Joint variables of the second stage are studied during the second part of the protocol, consisting in 3 subtle rotations of 1° each, around the $z - axis$.

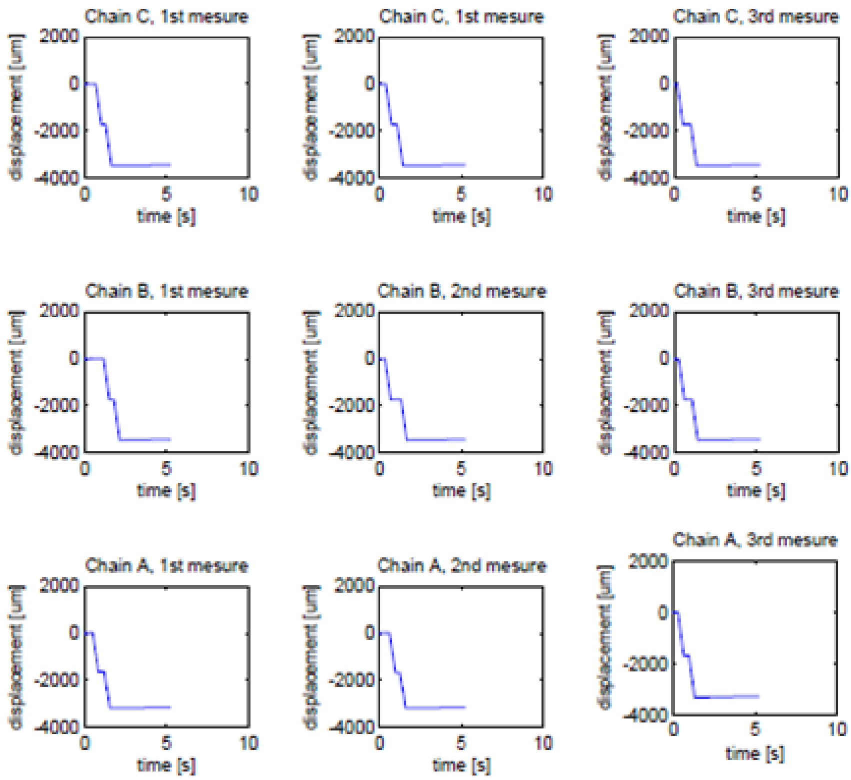


Figure 4.11: First stage actuators displacements acquired from interferometer: q_1 for Chain A, q_4 for chain B and q_7 for chain C are recovered

The main aspect to take into account when studying q_2 and q_5 is that their relative positioner's alignment with the interferometer's optical axis is affected by the simultaneous motion of the first stage, when the motion sequence is applied.
Signals recovered from interferometers are plotted on Matlab and shown in fig. 4.12. Three steps, corresponding to the three applied 1°-rotations, are clearly remarquable in the second part of each plot. In the first part, a subtle motion ($\tilde{}200nm$), shaped as another step, is evident too: it corresponds to the first part of the applied protocol, the translational one, in which the second stage should not move. The first stage's underlying joint variables experiment a displacement which originates a consequential motion also in the second stage.
Moreover, during the second part of the protocol, neither (q_1, q_4 and q_7), nor (q_2 and q_5) are equal to zero: consequently, the recorded outcomes for second stage joint variables could be slightly different in the experimental test if compared to the SA-IGM values. In addition, the alignment could suffer from this contemporary motion.
The width of the medium step is computed, as in the first stage case: here it corresponds to a 1° of rotation around z-axis. (see tab. 4.7)

Medium step	SA-IGM [mm]	SmarPod [mm]
Chain A	0.2787	0.314
Chain B	0.2787	0.305

Table 4.7: *Medium single step comparison, for an applied rotation of 1 °around z-axis, between SA-IGM and real SmarPod*

In plots, rotation steps' related to chains A and B evolve in opposite directions, while from SA-IGM model, q_2 and q_5 have the same sign: in fact, the configuration of the experimental set-up, notably the orientation of the SmarPod mounted on the antivibrational plane, could be an explanation of this apparent incongruity.
Each joint variable, in other terms the total displacement of each positioner, is evaluated as well: results are shown in tab. 4.9 below.

Joint variables	SA-IGM [mm]	SmarPod [mm]
q_2	0.8358	0.9159
q_5	0.8358	0.8812

Table 4.8: *Joint variables comparison, for an applied total rotation of 3 °around z-axis, between SA-IGM and real SmarPod*

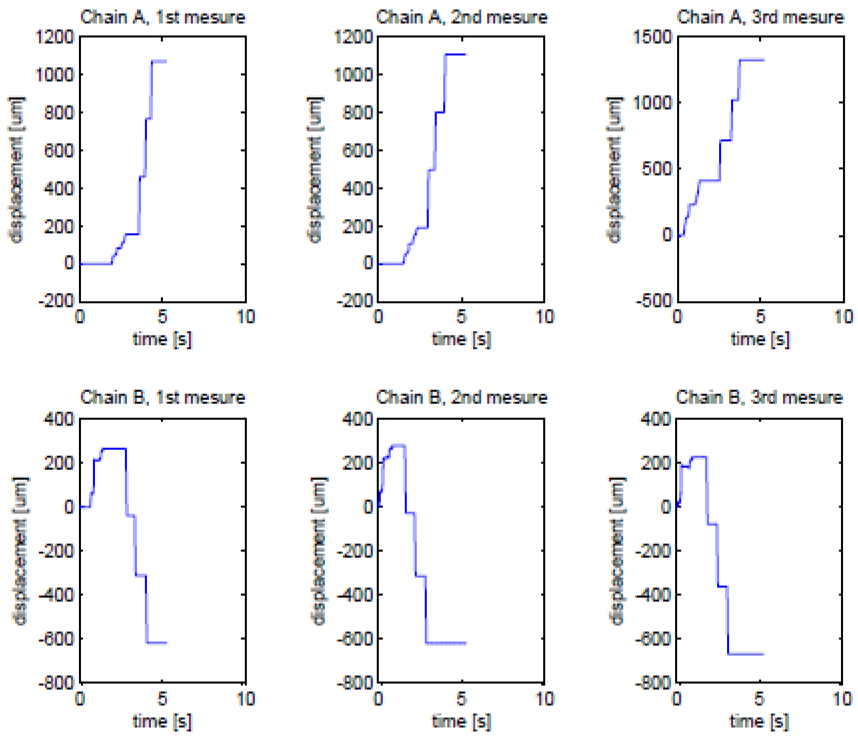


Figure 4.12: Second stage actuators displacements acquired from interferometer: q_2 for Chain A and q_5 for chain B are recovered

4.2.4 Discussion

For both first and second stage actuators, the experimental tests performed in a clean room on a microrobotic work cell have analyzed their displacements after a selected motion sequence of translations and rotations.
The final goal of the experimental validation was recovering **joint variables** from the real SmarPod using a microrobotic cell and compare them with the ones obtained through the SA-IGM, in order to assess the role of VR simulation.
Therefore, in fig. 4.13 comparisons between mean values of joint variables extracted from interferometer's signal (mediated through the three repetitions) are provided and those calculated with the SA-IGM. First (fig. 4.13(a)) and second stage's (fig. 4.13(b)) analysis are here decoupled to stay coherent to the approach adopted above.
Errors between SA-IGM and the experimental validation's mean values are computed. They express system's absolute accuracy for each joint variable value.

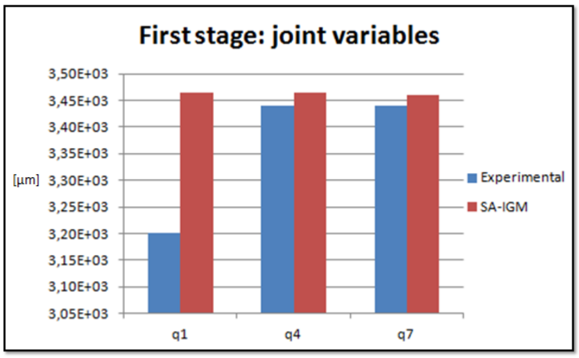
Joint variable	**Error** [μm]
q_1	263
q_4	24
q_7	20
q_2	80
q_5	54

Table 4.9: *Errors' computation between SA-IGM results and experimental validation ones*

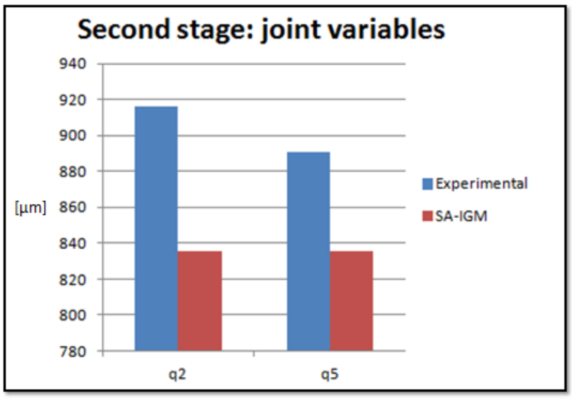
Joint variable	**Standard deviation** [μm]
q_1	61.57
q_4	0.107
q_7	0.13
q_2	0.027
q_5	7.86

Table 4.10: *Joint variables' standard deviations*

However, experimental conditions called for manual application of EE's displacements through SmarPod's interface: this must be taken into account when examining results. For example, the largest error measured corresponds to the first experimental measurements carried out and it exceeds 100 μm. Also q_1 values standard deviation is quite significant (61,57 μm). Errors concerning the others joint variables are all below this



(a) First stage joint variables' results comparison between SA-IGM and experimental measurements



(b) Second stage joint variables' results comparison between SA-IGM and experimental measurements

Figure 4.13: Joint variables' comparison between SA-IGM and experimental measurements

limit value and their standard deviation are acceptable. This can be explained with the operator's learning of the experimental procedure, in particular for what concerns the applications of motions through the interface and the robot-interferometer alignment condition. Larger errors extracted in the second stage (q_2, q_5) can be explained with the misalignment induced by the simultaneous motion of the first stage durin the first part of the sequence. The Simulation-Aided model can be considered validated and the role of VR is asserted: in fact, it has played a fundamental role in both modeling and validating the SmarPod's SA-IGM.

Conclusions and future developments

This thesis has presented a **non-conventional Inverse Geometrical Model** of a commercial micropositioner used for SEM applications, being vacuum-compatible. Its complex, 3-staged architecture and its parallel kinematics, along with its small dimensions (115 x 101.4 x 25 mm^3), call for an "ad-hoc" resolution, possibly aided by adequate tools. Dealing with parallel robots' modeling, in fact, it is not as straightforward as with the serial ones. The preliminar step to understand and represent robot's kinematics is to recover a Geometrical Model (GM) of the positioner itself. An accurate analysis of positioners' structure and techical features has been necessary before starting modeling its kinematic behaviour. Main focus of this project has been the development of the micropositioner's IGM: in other terms, the objective was to extract **actuators' displacements** (named joint variables, collected in vector $\overrightarrow{q}$) required to achieve a desired position and orientation of the moving stage of the robot (position and orientation information of an EE are collected into the pose vector $\overrightarrow{x}$).

Various solutions to parallel robot's IGM are provided in literature, most commercialized examples being fully parallel, which means that the number of legs of the robot is equal to the number of DoF of its moving stage. On the other hand, when a **not fully parallel robot** is involved, examples are more rare. SmarPod micropositioner presents three kinematic chains linked to a 6-DoF moving platform and, therefore, it is defined as a not fully parallel micromanipulator.

In order to proceed with a valid modelisation of the micropositioner, a typical approach generally employed for fully parallel robots has been adapted to this case study, assuming the system's **linearity** and thus splitting the problem into position and orientation. Therefore, the so-called **Analytical-IGM** (A-IGM) has been developed.

Then, a VR 3D-reconstruction of the micropositioner has been employed to discover its noteworthy features, not easily visible to the naked eye: a clear and magnified visualization of robot's architecture and motion patterns is provided by the 3D virtual model. Moreover, it allows an **intuitive interaction** between virtual environment and the user, thanks to communication between Pyhton scripts and virtual robots's logical workflow of sensors, controllers and actuators. Consequently, VR helps to address both typical microscale and SEM issues of lack of sensing and opens new possibilities in microrobotics scenario, such as accurate task-planning to be performed before concretly dealing with the instrumentation, enhancing tasks' and tools' safety, accuracy and repeatability.

In this scenario, VR Blender environment has played a double powerful role to this thesis' purposes. In the first instance, a new Python code has been developed: it allowed the simulation of simple motions applied on the moving stage of the virtual robot. Consequently, single contribution of each actuator to the global motion of the EE have been found and the so-called **Simulation-Aided IGM** has been developed. It provides an intuitive and efficient expression of SmarPod's joint variables, without resorting to traditional kinematic approaches. Then, virtual joint variables are collected too, thanks to the coupling between Blender *GameEngine* and the Pyhton code: thy will be employed as a **preliminar validation** tool assuming 3D model's validity.

The Blender preliminar validation has confirmed the SA-model to be the most accurate between the two: the final step of the followed workflow has been to find a possible experimental way to further assest the validity of the model and, with it, the role of Blender as a VR assistant to modelisation. For this purpose, a microrobotic workcell has been employed, taking advantage of interferometer's ability to detect position changes' in object aligned to its optical axis and to convert them into processable signals. The experimental validation provided promising results concering the SA-IGM's accuracy.

Possible **future developments** to complete and enlarge the present thesis work could be, in a first istance, avoiding resorting to interferometer, whose measurements are largely affected by vibrations, alignments' quality and experimental conditions. A solution should be extracting signals related to joint variables directly from actuator's encoders, therefore proceeding with a **closed-loop** analysis instead of an open-loop one.

In fact, selecting the interferometer as the tool to carry out the experimental validation in the microrobotic working cell has been useful to ensure SA-IGM to provide results coherent with the SmarPod's kinematic behaviour, taking advantage of the tool working principle. However, the experimental protocol has been applied manually to the robot, thanks to its dedicated PC interface: consequently, motions are not equally distributed in the available time interval. Moreover, **alignment quality**, especially for the second stage's joint variables, is afftected and slightly modified by the contemporary motion of the underlying first stage. Carrying out a closed-loop extraction of joint variables would surely result more robust and a direct comparison between joint variables' values would be possible.

The **DGM** could be addressed as well, taking advantage of the asserted role of Blender *GameEngine*. Without resorting to complex and time-consuming solutions proposed in literature, it could be implemented a dedicated strategy, such as the one developed in this thesis work, to recover EE's pose vector $\overrightarrow{x}$, once joint variables' vector $\overrightarrow{q}$ is given. Finally, differential kinematics and **dynamics** of this very particular and not fully parallel micropositioner could be explored as well, once the DGM and IGM are available and verified. Great interest in dynamics of Stick-Slip (SS) actuators has spread in the very last years: for example, in this case study, an innovative application could be a comparison between SS positioners' behaviour in standard conditions and that under SEM's vacuum. To achieve that, a complete and detailed study of literature concerning friction models would be needed, along with kinematics of parallel robot.

Appendix A

Analytical IGM

The workflow explained in Chapter 2 is explained in the following paragraphs. Thanks to a preliminar Blender analysis, positions of rotules A, B and C in the GRF have been recovered, as explained in Chapter 3. ABC triangle results to be equilateral (as shown in figure 14):
$\overrightarrow{AB} = \overrightarrow{BC} = \overrightarrow{CA} = d$, being triangle's sides;
$\overrightarrow{EA} = \overrightarrow{EB} = \overrightarrow{EC} = e$, being the radii of the circumscribed circumference;
$\delta = 30°$

The final aim is to obtain rotules' positions as functions of their relative joint variables, each in its local reference frame LA, LB or LC:

$$\overrightarrow{p_A^{LA}} = f(\overrightarrow{q_A});$$
$$\overrightarrow{p_B^{LB}} = f(\overrightarrow{q_B});$$
$$\overrightarrow{p_C^{LC}} = f(\overrightarrow{q_C}).$$

Chains are treated separately from each other (see fig. 15): resulting all identical, it's possible to analyze a single chain, for example chain C, and then extending results to the other ones.

The selected approach consists in calculating the homogenous matrix expressing rotules' coordinates and orientation in the LC reference frame. It results, considering non-fixed axes:

$$C^L = C_1^0(q_1)C_2^1(q_2)C_3^2(q_3);$$

However, having choses all LRF parallel to the GRF, the analysis results simplified as the rotation section of each homogeneous matrix results an identity for all chains:

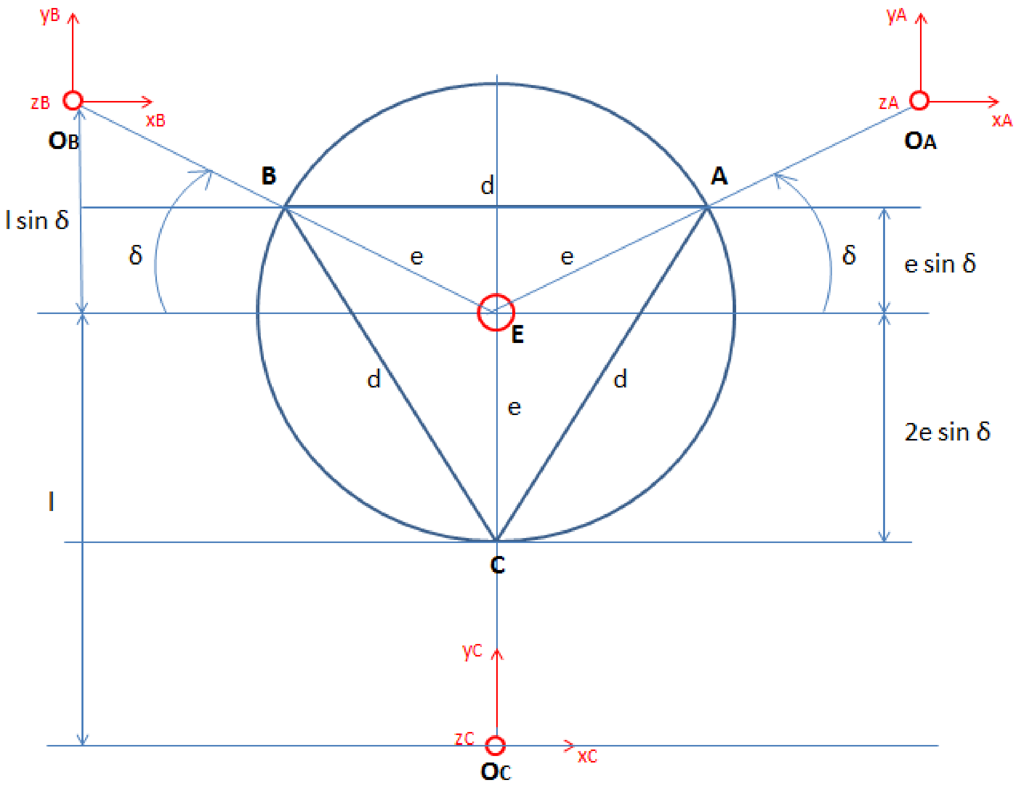


Figure 14: System's top view: in red Local Reference Frames (LRF), one for each leg. Global Reference Frame (GRF) lies on the xy plan, resulting parallel to the local ones: its origin O_E lies on the fixed base, in correspondance of platform's E point and translated along the z-axis of -16.71 [mm]

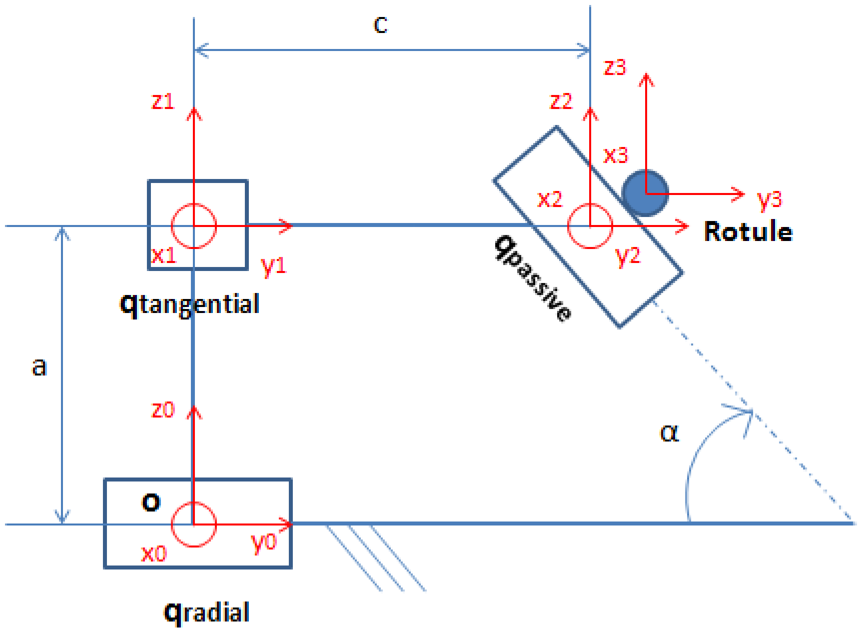


Figure 15: Single chain's modelisation (after "cut"): the Local Reference Frame (LRF) is the "0" one, centred in O

$$\left(\begin{array}{ccc|c} 1 & 0 & 0 & q_{tangential} \\ 0 & 1 & 0 & q_{radial} + c + q_{passive}cos\alpha \\ 0 & 0 & 1 & a - q_{passive}sin\alpha \\ \hline 0 & 0 & 0 & 1 \end{array}\right)$$

Expressions of rotules' position are now expressed in the GRF:

$$\overrightarrow{p_A^G} = \overrightarrow{p_A^{LA}} + \overrightarrow{o_A};$$
$$\overrightarrow{p_B^G} = \overrightarrow{p_B^{LB}} + \overrightarrow{o_B};;$$
$$\overrightarrow{p_C^G} = \overrightarrow{p_C^{LC}} + \overrightarrow{o_C};.$$

In fact, vectors $\overrightarrow{o_A}, \overrightarrow{o_B}$ and $\overrightarrow{o_C}$ are known by geometry:

$$\overrightarrow{o_A} = [lcos\delta; lsin\delta; 0; 1];$$
$$\overrightarrow{o_B} = [-lcos\delta; lsin\delta; 0; 1];$$
$$\overrightarrow{o_C} = [0; -l; 0; 1];$$

Closure equations are now imposed in the form:

$$\overrightarrow{p_A^G} = g(\overrightarrow{x_E});$$
$$\overrightarrow{p_B^G} = k(\overrightarrow{x_E});;$$
$$\overrightarrow{p_C^G} = w(\overrightarrow{x_E}),$$

in which x_E is the vector of the displacements applied to the EE: $\overrightarrow{x_E} = (x_E, y_E, z_E)$. In the end, it results:

$$\begin{cases} x_C = x_E; \\ y_C = y_E - 2esin\delta; \\ z_C = z_E - a; \end{cases}$$

$$\begin{cases} x_A = x_E + \frac{esin\delta}{2}.; \\ y_A = y_E + esin\delta; \\ z_A = z_E - a; \end{cases}$$

$$\begin{cases} x_B = x_E - \frac{esin\delta}{2}.; \\ y_B = y_E + esin\delta; \\ z_B = z_E - a; \end{cases}$$

Final systems of equations obtained to be implemented on Matlab:

$$\begin{cases} x_A = q_2 \\ y_A = q_3 cos\alpha + q_1 + c \\ z_A = a - q_3 sin\alpha \end{cases}$$

$$\begin{cases} x_B = q_5 \\ y_B = q_6 cos\alpha + q_4 + c \\ z_B = a - q_6 sin\alpha \end{cases}$$

$$\begin{cases} x_C = q_8 \\ y_C = q_9 cos\alpha + q_7 + c \\ z_C = a - q_9 sin\alpha \end{cases}$$

Simulation-Aided IGM

In this section it is shown how to extract the SA-IGM developed in Chapter 3. To do so, single contributions to the final expression of q_A, q_B and q_C will be explained. Two different cases of motions are considered while performing Blender's simulations:

- **Translation in the xy plan**: single contributions of each stage for translations on the $x - axis$ and on the y one are recovered. For chain A and chain B, the EE's displacement on this plane is splitted into two contribution, one for each of the first two stages, weighted by sinus and cosinus of angle δ. As far as chain C is concerned, on the other hand, an x-translation is transmitted to its second tangential actuator, while a translation along y is transmitted to its first radial stage. For all stages, the third passive-slide stage does not contribute to motions on the xy plan.

$$\textbf{X-axis translation} \begin{cases} q_1 = d_x cos\delta; \\ q_2 = d_x sin\delta; \\ q_3 = 0; \end{cases} \quad \begin{cases} q_4 = -d_x cos\delta; \\ q_5 = d_x sin\delta; \\ q_6 = 0; \end{cases} \quad \begin{cases} q_7 = 0; \\ q_8 = d_x; \\ q_9 = 0; \end{cases}$$

$$\textbf{Y-axis translation} \begin{cases} q_1 = d_y sin\delta; \\ q_2 = -d_y cos\delta; \\ q_3 = 0; \end{cases} \quad \begin{cases} q_4 = d_y sin\delta; \\ q_5 = d_y cos\delta; \\ q_6 = 0; \end{cases} \quad \begin{cases} q_7 = -d_y; \\ q_8 = 0; \\ q_9 = 0; \end{cases}$$

- **Translation along z-axis**: observing Blender's animations is possible to derive that, to achieve a translation along the z-axis, only the first and the third stage do move, while the second one stays fixed

$$\begin{cases} q_1 = d_z ctg\alpha; \\ q_2 = 0; \\ q_3 = d_z \frac{1}{sin\alpha}; \end{cases} \quad \begin{cases} q_4 = d_z ctg\alpha; \\ q_5 = 0; \\ q_6 = d_z \frac{1}{sin\alpha}; \end{cases} \quad \begin{cases} q_7 = d_z ctg\alpha; \\ q_8 = 0; \\ q_9 = d_z \frac{1}{sin\alpha}; \end{cases}$$

Appendix B

Python script

In this Appendix, the whole Python code developed on Blender is provided:

```
Python 3.4.2 (v3.4.2:ab2c023a9432, Oct  6 2014, 22:15:05) [MSC v.1600 32
    bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> #Each script begins with the inclusion of the necessary libraries and
    modules.
#In this case, for example, the module mathutils includes vectors, matrix
    and their associated calculations.
from bge import logic
from socket import *
from struct import unpack
import mathutils
import math

# Get the current scene
scene = logic.getCurrentScene()

# Get objects named plateau, rotuleA, etc from the current scene
plateau = scene.objects["Plateau"]
rotuleA = scene.objects["Rotule_A"]
rotuleB = scene.objects["Rotule_B"]
rotuleC = scene.objects["Rotule_C"]
niveauplateau = scene.objects["Niveau_plateau"]
echantillons = scene.objects["Echantillons"]
size = 1024

# Get armatures A, B and C from the scene
scene = logic.getCurrentScene()
armature_baseA = scene.objects["Armature_positionneur_A"]
armature_baseB = scene.objects["Armature_positionneur_B"]
armature_baseC = scene.objects["Armature_positionneur_C"]

# Get armatures' bones of interest to recover joint variables
os_base1 = armature_baseA.channels["Bone"]
os_base2 = armature_baseA.channels["Bone.004"]
```

```
os_base3 = armature_baseA.channels["Bone.006"]

os_base4 = armature_baseB.channels["Bone"]
os_base5 = armature_baseB.channels["Bone.002"]
os_base6 = armature_baseB.channels["Bone.004"]

os_base7 = armature_baseC.channels["Bone"]
os_base8 = armature_baseC.channels["Bone.002"]
os_base9 = armature_baseC.channels["Bone.004"]

#Define main to print rotules' positions to screen
def main():
    plateau['x'] = plateau.worldPosition[0]
    plateau['y'] = plateau.worldPosition[1]
    plateau['z'] = plateau.worldPosition[2]

    rotuleA['x'] = rotuleA.worldPosition[0]
    rotuleA['y'] = rotuleA.worldPosition[1]
    rotuleA['z'] = rotuleA.worldPosition[2]

    rotuleB['x'] = rotuleB.worldPosition[0]
    rotuleB['y'] = rotuleB.worldPosition[1]
    rotuleB['z'] = rotuleB.worldPosition[2]

    rotuleC['x'] = rotuleC.worldPosition[0]
    rotuleC['y'] = rotuleC.worldPosition[1]
    rotuleC['z'] = rotuleC.worldPosition[2]

# Extract joint variables as differences between bones' heads and tails
    vect7 = (os_base7.pose_head-os_base7.pose_tail).length
    vect8 = (os_base8.pose_head-os_base8.pose_tail).length
    vect9 = (os_base9.pose_head-os_base9.pose_tail).length
    vect4 = (os_base4.pose_head-os_base4.pose_tail).length
    vect5 = (os_base5.pose_head-os_base5.pose_tail).length
    vect6 = (os_base6.pose_head-os_base6.pose_tail).length
    vect1 = (os_base1.pose_head-os_base1.pose_tail).length
    vect2 = (os_base2.pose_head-os_base2.pose_tail).length
    vect3 = (os_base3.pose_head-os_base3.pose_tail).length

#Eliminate offset in joint variables' expressions
    if ( hasattr(logic, "init_q") == 0 ):
        logic.q1 = vect1
        logic.q2 = vect2
        logic.q3 = vect3
        logic.q4 = vect4
        logic.q5 = vect5
        logic.q6 = vect6
        logic.q7 = vect7
        logic.q8 = vect8
```

```
        logic.q9 = vect9
        logic.init_q = True

    print('q1= ', vect1 - logic.q1)
    print('q2= ', vect2 - logic.q2)
    print('q3= ', vect3 - logic.q3)
    print('q4= ', vect4 - logic.q4)
    print('q5= ', vect5 - logic.q5)
    print('q6= ', vect6 - logic.q6)
    print('q7= ', vect7 - logic.q7)
    print('q8= ', vect8 - logic.q8)
    print('q9= ', vect9 - logic.q9)

#     if ( hasattr(logic, "connection_translation_SmarPod") == 0 ):
#         logic.connection_translation_SmarPod = socket(AF_INET, SOCK_DGRAM)
#         logic.connection_translation_SmarPod.setblocking(0)
#         logic.connection_translation_SmarPod.bind( ('127.0.0.1', 1085) )
#
    try:
# Assign a defined orientation to the moving platform
        plateau.worldOrientation = [[1.0,0.0,0.0],[0.0,-1.0,0.0],[0.0,0.0,-1.0]]
        angle_x = 0*math.pi/180
        angle_y = 0*math.pi/180
        angle_z = 20*math.pi/180
        plateau.applyRotation([angle_x, angle_y, angle_z])

        print('echantillons', echantillons.worldPosition)

    except:
        print("ERROR translation smarpod")
        pass

    return ;
if __name__ == "__main__":
    main()
```

Appendix C

Preliminar Blender validation

In the following Appendix, various examples of comparisons between Blender-recovered virtual joint variables and those obtained through the two developed model (for Analytical Inverse Geometrical Model A-IGM see Chapter 2 and for Simulation-Aided Inverse Geometrical Model SA-IGM see Chapter 3)are provided. SA-IGM has been chosen through this preliminar analysis to be the most Blender coherent. Therefore, it has been experimentally validated using a microrobotic workstation which included the real SmarPod (see Chapter 4).

Translations

	q_A	q_B	q_c
A-IGM	[15.44; 17.51; 0]	[-15.95; 12.52; 0]	[-6.9 ;-4.5; 0]
SA-IGM	[-0.397; -8.31; 0]	[7.394; 3.812; 0]	[-6.93; -4.5; 0]
Blender	[-0.397;-8.32;0]	[7.397; 3.812;0]	[-7; -4.5; 0]

Table 11: Translation of the End-Effector (EE) along x = -4.5 mm, along y = 7 mm.

	q_A	q_B	q_c
A-IGM	[16.84; -5.5; 0]	[-17.36; 24.53; 0]	[-5.5; -7.5; 0]
SA-IGM	[9.242; -1.013; 0]	[-3.74; 8.51; 0]	[-5.5;-7.49; 0]
Blender	[9.245; -1.013; 0]	[-3.75; 8.51; 0]	[-7; -4.5; 0]

Table 12: Translation of the End-Effector EE along x = 7.5 mm, along y = 5.5 mm.

Rotations

	q_A	q_B	q_c
A-IGM	[15.35; -14.51; 2.12]	[-18.86; 15.53; 2.12]	[-7; -1.5; 2.12]
SA-IGM	[4.05; -5-51; 3]	[6.65; 4.01; 3]	[-2.9; -1.5; 3]
Blender	[4.05; -5.51; 3]	[6.65; 4.01; 3]	[-2.9; -1.5; 3]

Table 13: Translation of the EE along x = -1.5 mm, along y = 5.5 mm, along z = 15.21 mm.

	q_A	q_B	q_c
A-IGM	[20.83; -13.58; 0.36]	[-24.37; 12.94; -2.37]	[0.97; 3.84; 2.43]
SA-IGM	[-1.45; -1.38; 1.22]	[-4.98; -2.85; 3.61]	[3.16; 3.89; 2.98]
Blender	[-1.45; -1.38; 1.22]	[-4.98; -2.86; -3.6]	[3.16; 3.88; 2.98]

Table 14: Rotation of the EE around x = 5°, around y = 5°, around z = 10°.

	q_A	q_B	q_c
A-IGM	[25.78; -13.01; -1.68]	[-23.4; 13.01; 1.68]	[1.1; 0; 4.21]
SA-IGM	[3.18; 1.94; 2.4]	[-3.18; -1.94; 2.4]	[7.03; 0; 5.95]
Blender	[3.18; 1.94; 2.38]	[-3.18; -1.94; 2.38]	[7.03; 0; 5.95]

Table 15: Translation of the EE around x = 10°.

	q_A	q_B	q_c
A-IGM	[15.35;-14.51;2.12]	[-18.86; 15.53; 2.12]	[-7; -1.5; 2.12]
SA-IGM	[4.05; -5-51; 3]	[6.65; 4.01; 3]	[-2.9; -1.5; 3]
Blender	[4.05; -5.51; 3]	[6.65; 4.01; 3]	[-2.9; -1.5; 3]

Table 16: Translation of the EE around z = 30°.

Bibliography

Abrahamians, J. O., Sauvet, B., Polesel-Maris, J., Braive, R., Regnier, S. (2013). "Robotic in situ stiffness cartography of InP membranes by dynamic force sensing." *IEEE International Conference on Intelligent Robots and Systems*, 1016–1021.

Ammi, M., Ferreira, A. (2007). Robotic assisted micromanipulation system using virtual fixtures and metaphors. *Proceedings - IEEE International Conference on Robotics and Automation*, (April), 454–460.

Arcese, L., Fruchard, M., Ferreira, a. (2012). "Endovascular magnetically-guided robots: navigation modeling and optimization." *IEEE Trans. Biomedical Engineering*, 59(4), 977–987.

Becker, B. C., Maclachlan, R. a., Lobes, L. a., Hager, G. D., Riviere, C. N. (2013). Vision-based control of a handheld surgical micromanipulator with virtual fixtures. *IEEE Transactions on Robotics*, 29(3), 674–683.

Bohringer, K., Fearing, K., Goldberg, K. (2007). "Microassembly." *In Handbook of Industrial Robotics, Second Edition.*

Bolopion, A., Dahmen, C., Stolle, C., Haliyo, S., Régnier, S., Fatikow, S. (2012). Vision based Haptic Feedback for Remote Micromanipulation in a SEM Environment. *International Journal of Optomechatronics*, 6(3), 236–252.

Bolopion, A., Stolle, C., Tunnell, R., Haliyo, S., Régnier, S., Fatikow, S. (2011). Remote microscale teleoperation through virtual reality and haptic feedback. *IEEE International Conference on Intelligent Robots and Systems*, 894–900.

Cassier, C., Ferreira, a., Hirai, S. (2002). Combination of vision servoing techniques and VR-based simulation for semi-autonomous microassembly workstation. *Proceedings 2002 IEEE International Conference on Robotics and Automation* (Cat. No.02CH37292), 2(May), 1501–1506.

Cecil, J., Powell, D., Vasquez, D. (2007). Assembly and manipulation of micro devices-A state of the art survey. Robotics and Computer-Integrated Manufacturing, 23(5), 580–588.

Cui, L., Marchand, E., Haliyo, S. (2014). 6-DoF automatic micropositioning using photometric information, Advanced Intelligent Mechatronics (AIM), 0–5. doi:10.1109/AIM.2014.6878197, pp. 918-923

Dario, P., Valleggi, R., Carrozza, M. C., Montesi, M. C., Cocco, M. (1999). "Microactuators for microrobots: a critical survey. Journal of Micromechanics and Microengineering", 2(3), 141–157.

Demaghsi, H., Mirzajani, H., Ghavifekr, H., "A novel electrostatic based microgripper (cellgripper) integrated with contact sensor and equipped with vibrating system to release particles actively" *Journal Microsystem Technologies*, 20(12), pp.2191-2202, 2014

Eda, H., Ishikawa, T., Tomita, Y., Yamamoto, Y., Zhou, L., Kawakami, K., Shimizu, J. (2003). "Development of biological-micro-manipulation system in scanning electron microscope." MHS2003. *Proceedings of 2003 International Symposium on Micromechatronics and Human Science* (IEEE Cat. No.03TH8717), 1–4.

Fatikow, S., Benz, M. (1998). "A microrobot-based automated micromanipulation station for assembly of microsystems." *Computers in Industry*, 36(1-2), 155–162.

Fatikow, S., Mounassypov, R. (1998). "Microassembly planning for manufacturing by flexible microrobots." *Proceedings. 1998 IEEE International Conference on Robotics and Automation* (Cat. No.98CH36146)

Ferreira, A., Cassier, C., Hirai, S. (2004). "Automatic microassembly system assisted by vision servoing and virtual reality." *IEEE/ASME Transactions* on Mechatronics, 9(2), 321–333.

Ghanbari, A., "Force interaction and sensing in bio-manipulation", *A thesis submitted for partial fulfillment of the degree in Doctor og Philosophy in Mechatronics Engineering*, University of Canterbury, 2012

Ghanbari, A., Wang, W. W. W., Hann, C. E., Chase, J. G., Chen, X. C. X. (2009). "Cell image recognition and visual servo control for automated cell injection." *2009 4th International Conference on Autonomous Robots and Agents,* pp. 92–96.

Haliyo, D. S., Regnier, S. (2003). "Advanced applications using [mu]MAD, the adhesion based dynamic micro-manipulator." Proceedings 2003 *IEEE/ASME International Conference on Advanced Intelligent Mechatronics* (AIM 2003), 2(Aim).

Haliyo, D. S., Venture, G., Regnier, S. (2005). "Tele-manipulation by adhesion of micro objects." *2005 International Symposium on Computational Intelligence in Robotics and Automation*, 133–138.
Haliyo, S. (2003). Advanced micro-manipulation applications, *Intelligent Robots and Systems, 2003. (IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on (Volume:2)* (October), 3–8.

Kaminskyj, S. G. W., Dahms, T. E. S. (2008). "High spatial resolution surface imaging and analysis of fungal cells using SEM and AFM." *Micron*, 39(4), 349–361.

Krijnen, G., Tas, N. "Micromechanical Actuators" 1–41.

Krishnan, S., Saggere, L. (2012). "Design and development of a novel micro-clasp gripper for micromanipulation of complex-shaped objects." *Sensors and Actuators*, A: Physical, 176, 110–123.

Ladjal, H., Hanus, J.-L., Ferreira, A. (2011). "Microrobotic simulator for assisted biological cell injection." 2011 *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1315–1320.
Latorre, L., Kim, J., Lee, J., De Guzman, P. P., Lee, H. J., Nouet, P., Kim, C. J. (2002). "Electrostatic actuation of microscale liquid-metal droplets." *Journal of Microelectromechanical Systems*, 11(4), 302–308.

Latt, W. T., Tan, U. X., Georgiou, A., Sidarta, A. E., Riviere, C. N., Ang, W. T. (2012). "A micro motion sensing system for micromanipulation tasks." *Sensors and Actuators*, A: Physical, 173(1), 254–266.

Lee, H., Lee, S. a, Bergsneider, M., Judy, J. W. (2009). "Unobstructing magnetic microactuators for implantable catheters" *Solid-State Sensors, Actuators and Microsystems Conference, 2009. TRANSDUCERS 2009. International*, pp. 857–860.

Liu, D. K.-C., Friend, J., Yeo, L. (2010). "A brief review of actuation at the microscale using electrostatics, electromagnetics and piezoelectric ultrasonics." *Acoustical Science and Technology*, 31(2), 115–123.

Mattos, L. S., Dagnino, G., Becattini, G., Dellepiane, M., Caldwell, D. G. (2011). "A virtual scalpel system for computer-assisted laser microsurgery." *IEEE International Conference on Intelligent Robots and Systems*, (c), 1359–1365.

Merlet, J. (2001). "Parallel robots." Vasa.

Merlet, J.-P. (2001). "Micro parallel robot MIPS for medical applications." *ETFA 2001. 8th International Conference on Emerging Technologies and Factory Automation. Proceedings* (Cat. No.01TH8597), 2, 1–9.

Muthuswamy, J., Okandan, M., Jain, T., Gilletti, A. (2005). "Electrostatic microactuators for precise positioning of neural microelectrodes. IEEE Transactions on Biomedical Engineering", 52(10), 1748–1755.

Nakajima, M., Hirano, T., Kojima, M., Hisamoto, N., Homma, M., Fukuda, T. (2011). "Direct nano-injection method by nanoprobe insertion based on E-SEM nanorobotic manipulation under hybrid microscope." *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 4139–4144.

Nambi, M., Damani, A., Abbott, J. J. (2011). "Toward intuitive teleoperation of micro/nano-manipulators with piezoelectric stick-slip actuators." *IEEE International Conference on Intelligent Robots and Systems*, 445–450.

Nelson, B., J., Dong, L., Arai, F., "Micro/Nanorobots" (2008) in *Springer Handbook of Robotics*, ch. 18, pp. 411-450

Nelson, B. J. (1998). Detc98 / Mech-5836 a Virtual Reality Teleoperator Interface for Assembly of Hybrid Mems Prototypes. Design Engineering, 1–7.

Nelson, B. J., Zhou, Y., Vikramaditya, B. (1998). "Sensor-based microassembly of hybrid MEMS devices." *IEEE Control Systems Magazine*, 18(6), 35–45.

Ouyang, P. R., Zhang, W. J., Gupta, M. M., Zhao, W. (2007). "Overview of the development of a visual based automated bio-micromanipulation system." *Mechatronics*, 17(10), 578–588.

Peng, J. Y., Chen, X. B. (2010). "Short Papers Modeling of Piezoelectric-Driven Stick – Slip Actuators." *IEEE/ASME Transactions on Mechatronics*, 16(3), 1–6.

Riviere, C. N., Ang, W. T., Khosla, P. K. (2003). "Toward Active Tremor Canceling in Handheld Microsurgical Instruments." *IEEE Transactions on Robotics and Automation*, 19(5), 793–800.

Régnier, S., Chaillet, N., (2010)*Microrobotics for micromanipulation*, ISBN: 978-1-84821-186-5, Wiley-ISTE

Sauvet, B., Ouarti, N., Haliyo, S., Regnier, S. (2012). "Virtual reality backend for operator controlled nanomanipulation." 2012 *International Conference on Manipulation, Manufacturing and Measurement on the Nanoscale*, 3M-NANO 2012 - Conference Proceedings, 121–127.

Su, G. D. J., Chiu, C. W., Jiang, F. (2005). "Vertical micromirrors integrated with electromagnetic microactuators for two-dimensional optical matrix switches." *IEEE Photonics Technology Letters*, 17(9), 1860–1862.

Tam, E., Sausse Lhernould, M., Lambert, P., Delchambre, A., Delplancke-Ogletree, M. P. (2009). "Electrostatic forces in micromanipulation: Experimental characterization and simulation including roughness." *Applied Surface Science*, 255(18), 7898–7904.

Yu, S., Nelson, B. (2001). "Autonomous injection of biological cells using visual servoing." *Experimental Robotics VII*, 169–178.

Zhang, J., Yu, H., Gao, F., Zhao, X. (2011). "Key issues in studying parallel manipulators." *The 2011 International Conference on Advanced Mechatronic Systems*, 234–244.

For SmarPod 115.25 technical features: www.smaract.de

For Delta parallel robots: www.sutter.com

For Blender tutorial: Haliyo, Blender (2.5x) pour Simulation Physique

Ringraziamenti

Concludere questa tesi con dei *ringraziamenti*, per quanto il caldo e la stanchezza si facciano sentire adesso che il 17 luglio è ad un passo, è stato un pensiero in me costante da quando il 6 ottobre scorso ho iniziato, con adrenalina ed po' di incredulità, la mia avventura parigina. Vorrei dedicare un pensiero a tutte le meravigliose persone incontrate ed alle vecchie, salde conoscenze che mi hanno accompagnata in questi 9 mesi. Come tutti sapranno, il dono della sintesi non è il mio forte e, per evitare di allungare ulteriormente il brodo, sarà meglio cominciare.

Grazie a mia madre, per essere consigliera e amica, interprete dei miei pensieri e compagna di vita. Grazie per ispirarmi, ed avermi ispirata, ogni giorno. Grazie per avermi insegnato l'umiltà e la cura di me stessa e di chi mi sta intorno, per trasmettermi la tua forza silenziosa, uguale ed opposta alla mia, per riuscire a placare le mie ire facili e per ricordarmi, ogni tanto, di respirare. Grazie a mio padre, perché inconsapevolmente (o forse no) più cresco e più mi avvicino al suo essere. Grazie ad entrambi perchè sono fiera di ciò che sono e, se così è, lo devo soprattutto a voi. Grazie alla mia zietta preferita, per essere (di nuovo) qui con me in questo giorno importante, insieme al resto della truppa (Franci, Claudia, Massi). Zia, è da tanto che non mi proponi la tua domanda preferita!

Ringrazio la Prof. Menciassi per avermi concesso questa meravigliosa ed irripetibile opportunità, che per me ha rappresentato un punto di svolta, un'esperienza di formazione personale, professionale e, soprattutto, di vita. Grazie Prof. per avermi accompagnata in essa in modo costante, per la disponibilità dimostratami ed il tempo concessomi in ogni momento, dalla partenza al rientro in Italia, fino ad oggi. Il suo supporto durante il mio periodo francese è stato di grande valore per me, così come lo sono stati i suoi consigli in questo ultimo mese di scrittura.

Ringrazio Edoardo per la sua dedizione e la sua sincerità, grazie per il sostegno e la costante disponibilità che mi hai dimostrato in questi ultimi mesi, mi hanno aiutata ad affrontare al meglio il rientro in Italia, nonchè l'ultimo momento di terrore pre-discussione.

Merci Sinan, pour m'avoir accueillie dans ton équipe avec un grand, joyeaux sourire dès le premier jour. Je ne oublierai jamais le son de ton éclat de rire et, surtout, la patience et l'encouragement constant que tu m'as montré dans mes six mois de stage. Merci pour tout ce que tu m'as appris. J'espère qu'on pourra travailler bientot à nouveau ensemble.

Merci Stéphane, pour ta disponibilité, tes conseils et pour la passion que tu mets tout les

jours dans ton travaille, elle se voit bien. Merci Mokrane, Jean et Camille pour votre soutien technique et pour m'avoir appris des trucs franchement génials.
Grazie alle mie amiche-da-una-vita Chiara, Adriana, Francesca e Marina, ognuna a suo modo imprescindibile e meravigliosa. Grazie Chiara, perchè la nostra quotidianità, così come i nostri viaggi, reali e mentali, sono esilaranti e a a dir poco imprevedibili. Così come è imprevedibile il numero di anni da cui ci conosciamo. Quanti sono quindi?! Grazie Adriana, perchè ci siamo viste piccole pesti alla perenne ricerca di "perchè" (una di noi due in particolare, sì, tu), poi ragazzine alle prese con problemi insormontabili, attorno ad un tavolino giallo che, di sabato sera, sapeva pacificarli tutti. Adesso ci guardiamo, donne, ma mi piace pensare che i nostri occhi siano sempre quelli di quando giocavamo e chiacchieravamo senza sosta. Grazie Francesca, per avermi dimostrato che l'empatia esiste e che nè le circostanze nè gli eventi contingenti sono in grado di scalfirla. Marina, so quanto avresti voluto essere qui per goderti il pre, durante e post laurea, grazie perchè, anche se l'ultimo anno ci ha tenute separate, so di averti sempre al mio fianco, sempre pronta ad ascoltarmi, a consigliarmi e a spettegolare insieme a me. E anche grazie a Chiara e Marina, insieme, coinquiline storiche fin dalla folkloristica bettola di via Porta a Mare. Grazie per aver condiviso innumerevoli catastrofi casalinghe, elettrodomestici ammutinati, sessioni d'esame in pigiama ai limiti del trash, coronate da feste che la duchessa Charlotte si sogna soltanto, vicini da incubo e varicelle.
Ringrazio i miei Baluards, da 7 anni al mio fianco nell'affrontare quotidiane scemenze (Frufri e Vito), esami improbabili ("Simile ad Iliceto", griglie di avanzamento nelle ripetizioni multiple, ehm), biciclette smarrite ma mai dimenticate, viaggi immaginari (come dimenticare quello a Marrakech?!), persino scontri dal sapore epico (Arianna contro il flocculo). Grazie ad Iliceto, per essere "simile a Saracino", per le musiche orientaleggianti e i variopinti vestiti africani mai acquistati. Anche oggi, come tre anni fa, condividiamo lo stesso momento di gloria estemporanea e fine a se stessa. Auguri mio saggio e pittoresco Dottore. Se tu ed i tuoi timpani siete sopravvissuti 7 anni alla mia dolce e flebile vocina da zurlera, grandi cose vi aspettano! Grazie A.. F... Izzo, perchè in 7 anni sei riuscito a cambiare tagli di barba e capelli più di tutte le persone che conosco messe insieme! Grazie per la sopportazione (... è reciproca!), per la tua innata generosità e benevolenza (vedi Gallipoli estate 2014) e soprattutto per la mia nuova bici. Seriamente... (...) grazie perchè tu SBLAD, e lo sai, soprattutto quando ti vesti d'organza!
Grazie a Silvia, donna dal multiforme nomignolo, compagna di inenarrabili peripezie che spaziano geograficamente dalla terra delle Cave alla Lucanìa ed oltre, affascinante musa dai toni a dir poco acuti e dai fili d'oro abbaglianti... Tornando alla realtà, grazie Silvia per avermi fatto scoprire la raffinata nonchè delicatissima cucina del Nord Italia (a base di lardo, trippa, tordelli e testaroli), per aver puntellato le nostre giornate di interrogativi filosofici di alto livello (piumino non piumino, pizza o colonna?), per le nostre memorabili uscite culturali a due. Potrei continuare ad oltranza, e lo so che lo temi, perciò qui mi fermo. Grazie, Amica!

Grazie Parigi, che per sei mesi mi hai accolta nel tuo perenne scorrere fluido e protetta tra i tuoi boulevards incantati. Grazie ai tuoi angoli nascosti, così come a quelli più maestosi ed inevitabili, luoghi in cui tutto accade e può accadere, in cui gli occhi non riescono a fare a meno della Bellezza, ipnotica, che cola via rapida, ma subito pronta a rigenerarsi. Grazie per la chanson, per l'arte, per le baguettes fragranti ad ogni ora, per i marché aux fleurs, per i rari e splendidi soli che ti cambiavano la pelle, per il Grénier e l'Attirail, per l'Opera, per i derniers métro, per Belleville, per le meravigliose anime che mi hai regalato.
Grazie Francesca, ma cop, per essere stata esploratrice di un mondo nuovo al mio fianco, per le ore piene passate insieme, le cene su note jazz, le chiacchiere infinite. Grazie per i vin chauds alle Tuileries, per gli scalini di Notre-Dame, per esserti ritrovata insieme a me bambina sul manège, per le avventure nella gita londinese. Quando un incontro ha il sapore di predestinazione.
Grazie Marco, per essere stato la mia zona neutra.
Grazie Salvo (Valdo), mia guida fin dai primissimi giorni all'ISIR (quando ancora non sapevo cosa fosse il Deux Bis!), per le pause sigaretta/caffè/sigaretta tutte italiane, per i discorsi e gli sfoghi. Grazie perchè la passione in ciò che fai traspare tutta, dai tuoi occhi e dal tuo francese impeccabile (!), perchè il tuo entusiasmo è trascinante e perchè una buona parte delle mie riflessioni sul futuro, compresi cambiamenti di idea epocali, li hai ispirati tu. Jo Nesbo.
Ringrazio Mara, per la sua stralunata, romanaccia allegria (... ma Mara?!?) e Matteo, per le risate complici, alticce e musicali che abbiamo condiviso. Ah! Non si è più giocato a Barbuto!
Merci à vous, les amis "parisiens", vous avez croisé mon petit chemin, oui c'est vous, les amis du labo, doctorants et stagiaires de l'ISIR! Merci à toi Alex, mardito y amigo, bon, ça ira bien en italien aussi. Grazie per le innumerevoli "pause clopes", accompagnati dal tipico tempo di m... parigino e dalle altrettanto tipiche nostre figure di m. Grazie per la tua verità e spontaneità disarmanti e per la tua risata (ça me semble de la ressentir encore, certaines fois, en Italie!).
Merci aux seules, VRAIES parisiennes, Ninon et Camille, mes chouchous, pour m'avoir fait decouvir la beauté d'une ville comme... Brescia! Je rigole, merci les filles pour votre bonheur, toujours en souriant, soit au labo, soit dans une terrasse parisienne (trop stylé!). Merci pour m'avoir fait sentir comme si j'étais chez moi!
Merci Carlos (PIU, PIU!) pour l'amour que tu as montré pour la culture et la langue italienne, meme si les résultats n'étaient pas trop brillants... Je suis toujours prete à t'apprendre des phrases pour draguer!
Merci Basil (Baz), pour l'esprit libre qui se révèle de tes yeux et pour avoir, quand tu ne te trompais pas sur la grammaire, légérement amelioré mon français. T'inquiète pas, la rrréalité virrrrtouelle, elle est encore là! Merci Antoine, pour les leçons d'anglais les plus amusantes que j'aille jamais donné dans ma vie (et pour m'avoir appris, avec Laetitia, où

se trouve l'ile de la Réunion!). Merci Jordan pour m'avoir influencée avec ton français très "personalisé" et pour la soirée musique chez toi. Merci à tous les autres gens du labo, Laetita, les Valentins, Anis, Hanan, Ryan, Carlos, Arthur, Justine, parce que la mienne à été une experience super grace à vous!
Grazie Ludi, per la tua (...direi proverbiale) testardaggine, la tua forza e i tuoi saggi consigli; grazie perchè abbiamo condiviso viaggi, momenti di sclero (vedi questo intero ultimo mese pre-laurea), idiozie varie (sono sempre una mega-zanzara), cibo (soprattutto brioches, NB col pallino!) e interessi. E due, auguri Dottoressa! Riuscirsi a laureare per ben due volte il 17 luglio non è da pochi! Solo la tua passione per la simmetria poteva arrivare a tanto.
Grazie Francesca (scucci scucci) per il sostegno fisico e morale negli ultimi giorni di fuoco pre-scadenze, colonizzare un BioRobotics Institute deserto senza di te non sarebbe stato degno di nota! Grazie perchè anche se i nostri percorsi si sono incrociati tardi, ho subito trovato in te quasi uno specchio di me stessa.
Grazie Ghila per i flash incredibili che abbiamo condiviso viaggiando tra Parigi e Londra sul nostro personalissimo asse di conquista, grazie per avermi fatto scoprire Shoreditch e per avermi coccolata con delle english breakfast indimenticabili!
Grazie Simona, per il tuo stesso essere, per portarmi un sorriso e un "picciò" quando più ne ho bisogno, per la tua spontaneità e la tua curiosità disarmanti.
Grazie Andrea per essere stato punto fisso e imprescindibile di questi anni pisani, per tutte le canzoni, i libri, i film e le parole che ci siamo scambiati, per la complicità unica che ci ha uniti e ci unisce.
Grazie Ila, per essere stata compagna di confessioni segretissime nel nostro luogo del cuore (il romantico arco di dei Bagni di Nerone), per i ritorni in bicicletta, decisamente allo sbaraglio, e per avermi insegnato "Iemmc a fa na tazza!".
Grazie Lover, per le tue missioni aumenta-autostima (SIIIIIII) che arrivano sempre al momento giusto, sicuramente tuo malgrado, per gli incontri all'improvviso e le nostre intellettuali conversazioni in inglese, di cui entrambi conserviamo sempre molti ricordi.
Grazie Frankie per la tua dolcezza, i tuoi pezzi al piano che mi fanno sognare e per avermi fatto scoprire il panino con la scamerita al Mescitino (50 % grazie a te 50 % grazie a Lover, ok).
Grazie ai Baui (Marco, Nicoletta, Simona, Silvia, Frufri, Iliceto, Simona, Ghila, Chiara), per le conversazioni non-sense e la comicità demenziale che ci accomuna dal primo all'ultimo, a quando la prossima cena?!
Grazie a Sofia e Virginia, perchè ho una quantità indefinita di ricordi felici collegati a voi, perchè mi conoscete come le mie tasche e perchè, nonostante le strade che percorriamo adesso siano parallele, so di poter sempre contare su di voi.
Grazie ad Alejandro, Alice ed Alessandra che hanno allargato la nostra esclusivissima cerchia, allietando l'ultimo anno di corsi.
Grazie Emilio (auguri Dottore, e 3!) per i momenti di ansia, sclero e disagio condivise

in questo ultimo mese, tra lauree e domande di Phd, varie ed eventuali. E' finita! Ci siamo! Finalmente possiamo godercela un po'. Grazie a Rossellina per le nostre serate di confessioni e cibo (sogno ancora le Seadas!), per aver messo in risalto spesso e volentieri la mia anima danzereccia e per i nostri specialissimi addominali-risate in palestra, che mi mancano un bel po'.
Grazie a Giada, Alessandro e Marco, per i consigli, i suggerimenti e le chiacchiere condivise in questi ultimi due mesi a Pontedera. Bravih!
Grazie a tutti i tesisti, Matteo, Silvia, Maura (ti intrufolo qui!), Costantyna, Clementina, Irene, Giulia, Francesca, Simona, Martina, Ferdinando, Aliria per i pranzi succulenti che abbiamo condiviso, che se non altro in mia assenza eviteranno di protrarsi per lunghe ore ("Arianna, mangia!!!" (cit.)) e per le chiacchiere sulle nostre scalinate, o sui divanetti (i divanetti!). Uno speciale grazie a Matteo per avermi sostenuta nei miei trip non-sense e per aver inventato Maturotti.

Peace, people.
Vi voglio bene.
Arianna