

École polytechnique de Louvain

Analysis of the spine through a multibody model and IMU technology

Author: **Eve LEROY**
Supervisors: **Paul FISETTE, Gabriel ABEDRABBO**
Reader: **Greet KERCKHOFS**
Academic year 2018–2019
Master [120] in Electro-mechanical Engineering

The biomechanical analysis of the spine can give important information to clinicians so they can provide appropriate treatment to their patients. It is especially helpful for the surgical planning of people suffering from spine-related pathologies like scoliosis or lower back pain. One of the inputs of the models being used in biomechanical analyses are the kinematics of the spine. This is a highly complex system so in order to recover them, the most widespread technique used is optoelectric motion systems composed of cameras and markers affixed to the skin. But this technique can only be operated in specific laboratories, and are thus constraining for the patient and the clinicians. Moreover, this technique is quite expensive, so a new cheaper and more portable way of obtaining the kinematics of the spine needs to be found.

The aim of this master thesis is therefore to simplify the protocol of data acquisition. For this reason, the kinematics will be evaluated thanks to Inertial Measurement Units affixed on the pelvis and the thorax. Those sensors measure the acceleration, angular speed, and magnetic field of the body they are attached to. In order to assess the intervertebral efforts of the spine, a multibody model is developed, based on the anthropometric data of the patient and its spine geometry, as well as the kinematics of the spine and the pelvis like stated before, and the muscles surrounding the spine.

It will be demonstrated in this thesis that patient-specific models are important in order to obtain precise results for intervertebral efforts. The benefit of adding muscle structures to the skeletal model will also be presented. Daily life movements, such as walking, kneeling and jumping, will be performed in order for the IMU sensors to gather necessary data to compute the internal efforts of the vertebrae which will be juxtaposed together and compared in order to determine the most demanding movements.

Perspectives and improvements will also be proposed for future researches.

ACKNOWLEDGMENTS

Throughout this thesis, I have received a great deal of support and assistance.

I would first like to thank my supervisors P. Fisette and G. Abedrabbo for their guidance through each stage of the process. I would also like to acknowledge G. Abedrabbo who was always open whenever I ran into a trouble spot or had a question about my research and for his proofreading of this document.

My research would have been impossible without the aid and support of Professor Julien Lebleu, who did his best to provide me the necessary materials and information on the sensors used in this master thesis.

Heartfelt thanks go to my fellow former students Denis Deneire and Pierre-Yves Legros, who wrote the previous thesis on the subject and gave me valuable information about it.

A special thank you to Professor G. Kerckhofs who accepted to be part of my jury and kindly read this thesis.

I would also like to acknowledge my mother as the second proofreader of this thesis, and I am gratefully indebted to her for her very valuable comments.

I am profoundly grateful to my godparent André Smet who helped me throughout my cursus with dedication and wisdom.

Finally, I must express my very profound gratitude to my parents and grandmother for their unfailing support and continuous encouragement throughout my years of study. This accomplishment would not have been possible without them. And to all of you who contributed in some way to this thesis and helped me achieve my goals, a warm “Thank you”.

Eve Leroy

1.1	Schematic of the spine	1
1.2	Main back movements	2
1.3	Structures of systems	5
1.4	Elementary joints	6
1.5	Various methods to obtain Euler angles	8
1.6	Gradient descent algorithm: visualization	9
1.7	Elemental rotations for the ZYX sequence of Euler angles	9
2.1	Spine kinematics	13
2.2	Multibody model of the spine and IMUs (R=rotations, T=translations)	13
2.3	Spine model	15
2.4	Anatomical planes and axes	17
2.5	Frames in the visualization tool of Robotran	17
2.6	Anatomy terms of location	18
2.7	Location of the CM for the additional body	18
2.8	IMU model	20
2.9	Missing link between the spine model and the IMU model	21
2.10	Schematic of the different frames used in this model	21
2.11	Entire model	22
2.12	Direct dynamics approach (left). Inverse dynamics approach (right).	26
2.13	Model of the muscles	26
2.14	Maxwell model	27
2.15	Series elastic actuator	27
2.16	Computer implementation	28
2.17	Magnetic peak in the magnetometer data of the IMU	29
2.18	Wrapped (left) and unwrapped (right) Euler angles	29
2.19	Frequency spectrum of the kinematics data	30
2.20	Z-acceleration (and zoom) of the pelvis IMU for a forward flexion movement	30
2.21	Structure of the global variables	31
2.22	Cubic spline interpolation visualization	33
3.1	Drift	35
3.2	Delay	36
3.3	Start of the experiment: orientation of the x-IMU	36
3.4	Start of the experiment: setting of the x-IMU	36
3.5	Euler angles comparison (Z-axis)	37
3.6	Euler angles comparison (X-axis)	38
3.7	Noise	40
3.8	Pelvis change	42

3.9	Upper body change	42
3.10	Lumbar spine change	43
3.11	Simu 3 vs 4	44
3.12	Analogy of a block on a slope	45
3.13	Forces in the S1-L5 intervertebral disk for a forward flexion movement	46
3.14	Force in T1 in the S1-L5 intervertebral disk for a forward flexion movement (zoom)	47
3.15	Roll of the lower thorax IMU : input and output of the spline function	47
3.16	Roll of the pelvis IMU (after derspline function) : angle, angular speed, angular acceleration	48
3.17	Spring forces and zoom (500Hz noise)	48
3.18	Velcro band with x-IMU in it	50
3.19	Structure to tare the IMUs	50
3.20	Magnetic device	50
3.21	Positioning of the IMUs	51
3.22	Kinematics and dynamics of the S1-L5 intervertebral disk for a forward flexion	52
3.23	Kinematics and dynamics of the S1-L5 intervertebral disk for 2 jumps	52
3.24	Kinematics and dynamics of the S1-L5 intervertebral disk for 3 steps (walk)	53
3.25	Dynamics of the S1-L5 intervertebral disk for left and right flexions of the trunk	55
3.26	Dynamics of the S1-L5 intervertebral disk for lateral left and right rotations of the trunk	55
3.27	Efforts in the S1-L5 intervertebral disk for the kneeling movement	56
A.1	Calibration axis	60
A.2	Axis of rotation	60
A.3	1st semester organization	61
A.4	2d semester organization: part 1	62
A.5	2d semester organization: part 2	63

LIST OF TABLES

2.1	Stiffnesses of the lumbar intervertebral disks	14
2.2	Percentage of the different segments of the female human body with respect to its total mass and their mass for a 56 kg female human body.	16
2.3	Mass of the bodies of interest for a 56 kg female human body	16
2.4	Longitudinal length of different segments of the female human body, center of mass position, and sagittal, transverse and longitudinal radii of gyration.	18
2.5	Inertia of the bodies of interest (with respect to their CM) for a 56 kg female human body	19
2.6	Inertia of the bodies of interest (with respect to their CM) after the change: additional body (x10), lumbar vertebra (x50) and pelvis (x10)	20
2.7	Values expected and obtained for the angles between TB1 and the thorax	23
2.8	Values obtained for the IMU Euler angles and the angles between TB2 and the IMU . . .	24
2.9	Values expected and obtained for the final vector	24
2.10	Potential cutoff frequencies for different movements	30
3.1	Errors for the drift : IMU mode (left) and AHRS mode (right)	38
3.2	Delay for the different algorithms	39
3.3	Inertia of the bodies of interest (with respect to their CM) for a 56 kg female human body	41
3.4	Error rate for the flexion movement	52
3.5	Error rate for the two jumps	53
3.6	Error rate for the walking movement	53
3.7	Minimal and maximal values for the intervertebral (S1-L5) efforts of different daily life movements	54

Abstract	i
Acknowledgements	iii
List of Figures	v
List of Tables	vi
1 Introduction	1
1.1 Context	1
1.2 Biomechanics of the spine	1
1.3 Application of IMU technology to the field of biomechanics	6
1.4 Discussion of the state-of-the-art	11
1.5 Objectives	12
2 Modeling	13
2.1 Spine model	15
2.2 Integration of the IMUs in the multibody spine model	20
2.3 Simulation steps	21
2.4 Computer implementation	27
3 Experiments	35
3.1 Choice of the x-IMU mode/algorithm	35
3.2 Inertia effect on intervertebral disks efforts	41
3.3 Thorax driven by IMU vs thorax driven by muscles	44
3.4 Analysis of the noise	46
3.5 Rigid thorax hypothesis validation and intervertebral efforts of daily life movements	50
4 Conclusions and perspectives	57
A Project management	59
B Codes	65
B.1 Matlab code: synchronization	65
B.2 Matlab code: inertia calculation	67
B.3 C code: Butterworth implementation	68
B.4 C code: derspline	69
B.5 C code: spline	71

1.1 Context

Many pathologies, like lower back pain (LBP) or scoliosis, affect the patients in their daily life movements. In order to analyze and understand the mechanics behind certain pathologies, the need of biomechanical spine models is constantly growing. Indeed, the clinical field could use those models in order to apply an appropriate treatment to the spine of the patient or enhance rehabilitation tools.

Many different systems performing kinetic data acquisition are available on the market, the optical motion systems being the more widespread (Vicon) but other systems such as electromagnetic (FasTrak), ultrasonic (Zabris) or inertial (xSense, x-IMU) systems can be used, too.

This master thesis will focus on the analysis of the lumbar intervertebral disks efforts assessed via a multibody model and IMU technology. As a matter of fact, IMUs are affordable and portable, therefore free of all laboratories constraints unlike optic systems.

1.2 Biomechanics of the spine

The Oxford English Dictionary [1] defines biomechanics as “*the study of the action of external and internal forces on the living body, especially on the skeletal system*”. In order to study that, the spine anatomy needs to be understood, the spine geometry and motion recovered, and modeling tools assessed.

1.2.1 Spine anatomy

It is important to have a firm understanding of the anatomy of the spine because, indeed, this knowledge provides modelers crucial data that they can exploit to build mathematical models as close as possible to the reality. Therefore, it can help the practitioner establish a suitable clinical diagnosis and treatment of a patient.

The spine, also called the vertebral column, consists of thirty-three vertebrae: seven cervical, twelve thoracic, five lumbar, five sacral and four coccygeal. As can be seen in FIGURE 1.1, the five sacral vertebrae are fused together. The same can be observed for the coccygeal vertebrae. Intervertebral discs are separating the upper twenty-four from each other allowing relative motions between vertebrae.

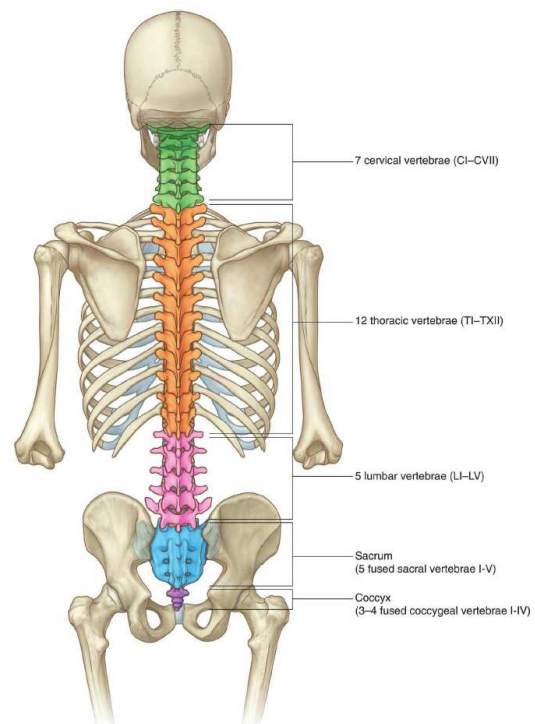


Figure 1.1: Schematic of the spine [2]

The back of a human being is more complex than an ensemble of vertebrae and intervertebral discs. In fact, the spine is attached to muscles interconnecting it to the ribs, the pelvis and the skull. Moreover, the back encloses the spinal cord which leads us to the major roles of the vertebral column:

- It protects the spinal cord and provides a safe passage for spinal nerves;
- It supports the weight of the upper body with the help of muscular elements of the back and favors an upright posture;
- It allows flexibility and mobility of the upper body. The main back movements (FIGURE 1.2) are the extension and flexion of the trunk, its lateral flexion and rotation.

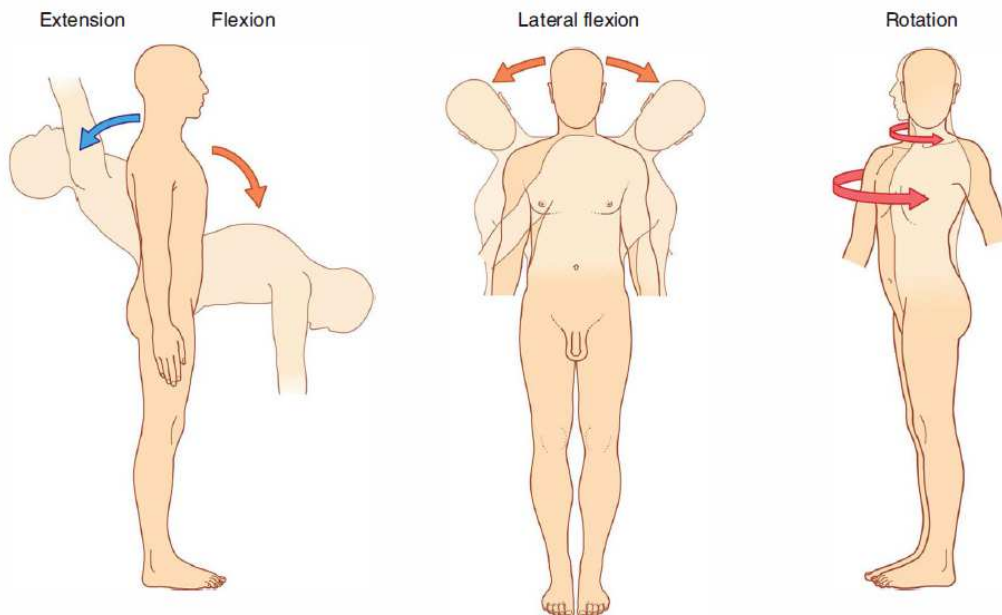


Figure 1.2: Main back movements [2]

The size and shape of the vertebrae vary depending on which region is examined. The same can be said for the mobility and function of the spine [3] [4] [5]:

- The cervical region is designed for mobility since it positions with precision the head and therefore the line of vision. At the same time, it must be sufficiently strong to carry the head;
- The thoracic region is the least mobile and the longest area of the spine. Its most important function is to protect the heart, lungs and major vessels thanks to the ribs connected to it. However, this protection comes at the expense of the mobility;
- The lumbar region supports the head, neck, trunk and upper limbs and transmits the forces and weight through the pelvis to the lower limbs. Moreover, it allows movement between the thorax and pelvis. The structure of this region optimizes mobility and stability thanks to the deformability of the intervertebral discs.

1.2.2 Spine geometry

R. Dumas [6] argues that a precise 3D representation of the spine is important in biomechanics because, first, it can help a surgeon to perform a diagnosis and surgical planning. Second, it is useful for engineers to build a personalized model for biomechanical studies but also for computer-assisted surgery. Indeed, S. Delorme [7] insists on the fact that if models aim at being used in a clinical framework, they have to

be patient-specific because each patient has a different spine geometry. They also state that personalized geometrical models have already been built in order to assess the effects of treatments.

There are two different ways of assessing the spine geometry, either in an indirect way or in a direct way. The indirect way states that the shape of the spine can be approximated by the shape of the skin surface. Its main advantage is that it is a non-invasive method. But taking skin measurements induces many unknowns, therefore a lack of accuracy appears. Indirect measurement tools include non-exhaustively skin markers, inclinometers, and optoelectric systems. In contrast, the direct way which is mainly based on radiographs has the advantage of assessing correctly the location of the vertebrae, but is more invasive due to the use of radiations. Direct measurement tools encompass computed tomography X-rays, plain and bi-planar radiology.

The basic mechanism of X-rays is that the thicker a material is, the more it absorbs. An other important parameter of the matter is the density of it. Therefore, denser tissues like bones absorb more X-rays than soft tissue. Radiographs give the absorption rate of the material since air ($\approx$ no absorption) will be represented in black on the radiograph and metal ($\approx$ full absorption) in white.

Although numerous ways of obtaining the geometry of the spine exist, G. Abedrabbo [8] chose a 3D reconstruction from bi-planar radiographs. V. Pomero [9] argues that *“Stereoradiographic protocols consisting in bi-planar radiography of different incidences, i.e., frontal and lateral standard radiographs, have been largely used in research environments as an alternative for 3D-reconstruction of the scoliotic spine in standing position”*. G. Abedrabbo states that some other approaches using the laying position (e.g. IRM) for the acquisition of the geometry of the spine, were not representing accurately the geometry of the spine when standing or walking since the lying and standing geometry were different. The other advantage of bi-planar radiographies is that it is a low cost solution since most hospitals have it. But the quality and precision of the images were lower compared to those of the EOS biplanar X-ray imaging technique consisting in taking simultaneously 2D anteroposterior and lateral images of the body in order to reconstruct a 3D model based on statistical models [10]. The EOS imaging technique was not used due to its price [8]. An other advantage of the bi-planar X-rays is that it exposes the patient to less radiations than X-ray computerized tomography (CT) and is more cost effective than magnetic resonance imaging (MRI) especially when metal implants are involved since MRI cannot be used anymore and the quality of the CT scans images become too low [11].

S. Kadoury [12] developed a self-calibration algorithm in order to reconstruct the 3D geometry of the spine. This reconstruction offered a versatile and robust for the 3D analysis of the spine in healthcare centers on the basis of standard uncalibrated radiographic images. Similarly, R. Pietton [11] also used a self-calibration algorithm that did not require a calibration tool in order to reconstruct the geometry of the spine. It used the natural anatomical landmarks of the spine and possible surgical implants. R. Dumas [6] developed a calibration method and a calibration device, basing himself on a simplified environment and a priori information to reconstruct the spine. The calibration device is a turntable to position the patient laterally or in the direction of the X-rays. G. Abedrabbo [8] also decided to build a rotary tray forcing the patient to stand still between the sagittal and coronal radiographies in order to obtain a good correlation between the two. He also used a calibration frame in opposition to both Kadoury and Pietton.

One of the spine models created by G. Abedrabbo was given as a starting point for this thesis.

1.2.3 Spine motion

The range of motion, or ROM, is *“the linear or angular distance that a moving object may normally travel while properly attached to another”* [13] or more simply *“the full movement potential of a joint, usually its range of flexion and extension”* [14].

W.-K Chiou [15] explains that the different spinal pathologies are associated to limited intersegmental mobility (ISM), the spinal motion being therefore affected by those disorders. They add that when an

injury to an intervertebral joint occurs, it alters the spinal mobility. It is therefore important to determine the individual vertebral contribution to the whole lumbar motion in order to understand the pathologies better. Y.-H Lee [16], who worked with Chiou, adds that one of the most common spinal disorders is low back pain (LBP). This pathology is the cause for a lot of the orthopedic surgeons and neurosurgeons visits.

A. Rozumalski [17] states that billions of dollars are devoted to lumbar spine pathologies each year, and even though the symptoms of these pathologies are well known, there is still problems in understanding the mechanical changes in the spine. Those mechanics have been previously assessed in vitro. P. Caravaggi [18] realized a bending experiment on cadavers in order to understand the effects of the intervertebral motion on cervical facet fractures.

Nowadays, in vivo methods are more widely used. There are two types of measures: discrete and continuous measures.

Discrete measures are conducted usually with goniometers and inclinometers (affixed on skin surface) but the latter measures spinal movements only in the sagittal plane, therefore lateral bending measurements being impossible to obtain [19]. The data acquisition process is difficult to automate due to the analogue nature of inclinometers [15]. In the experiment conducted by Y.-H Lee [16], lateral spinal radiographs were taken in the neutral, flexion, and extension movement. Radiology is also a standard technique for measuring the spinal motion but it has major limitations: it assess the spinal range only in one dimension, only allows specific postures and not a continuous motion, and requires high levels of X-rays. Y.-H Lee also talks about the cineradiography technique which shows the continuous motion of the spine, but releases even higher doses of radiation than classical X-rays.

Thanks to the development of 3D motion systems, continuous movement can be assessed via optoelectric motion analysis systems providing measurement of a patient's active range of motion. W.-K Chiou [15] cites the VICON, CODA-E-3, Spine Trak scanners as such devices. In the study, they compare the motion of intervertebral lumbar disks using both radiographic (invasive) and videographic (non-invasive) techniques. The skin markers coordinates were translated to vertebral positions thanks to non-linear mathematical models. M.A. Esola [20] conducted an experiment with a 3D optoelectric motion analysis system to assess the differences in motion in subjects having a low back pain (LBP) disorder. An other dynamical motion analysis device called electrogoniometer can be used to assess lumbar motion too [21]. They also use skin fixations and employ electromagnetic or potentiometric principles. M. Tojima [22] conducted an experiment comparing the VICON system with an electrogoniometer and found out that there was no difference in the results, both systems proving to be repeatable and reliable.

All of those methods, except radiographs are non-invasive. A. Rozumalski [17] states that *"few studies have directly measured in vivo lumbar spine kinematics with in-dwelling bone pins"*. This is an invasive method where, under anesthesia, wires are inserted into each vertebra. Cameras were used to assess the motion of the markers attached to those wires.

It was already stated that a lot of those instruments are affixed on the skin, but it is known that the motion of the skin does not reflect perfectly the motion of the underlying vertebrae. W.-K Chiou's experiment [15] was realized in order determine a relation between skin surface motion and intervertebral motion. Y.-H Lee [16] built a model transforming the skin markers into vertebrae positions.

1.2.4 Modeling tools

Two main techniques exist in order to model the human spine: finite elements (FE) and multibody models.

M. Lee [23] created a FE spine model in order to predict vertebral displacements resulting from lumbar manipulative forces, using short beam elements to represent the intervertebral disks. D. Coombs [24] developed a process to convert a CT scan of the lumbar spine into a model, by creating a mesh and assigning certain properties for each element. The model was solved using Abaqus [25], a widely spread FE modeling tool in the life science domain and computational biomechanics giving the engineer

materials, procedures, load types and implants information in order to build the model [26]. M. Rao [26] also used Abaqus in order to evaluate spine mechanics of the L2-L3 and L4-L5 intervertebral disks. H. Jaramillo Suárez [27] analyzed the stress and strain distribution in the L4-L5 and L5-S1 intervertebral disks. They used the CATIA and SOLIDWORKS programs (CAD programs) to build and assemble the vertebrae and the Abaqus software to analyze the model.

Finite element method Finite element modeling subdivides a large problem into smaller, simpler parts that are called finite elements. The simple equations that model these finite elements are then assembled into a larger system of equations that model the entire problem. Each element interacts with each other through nodal points. One of the most important advantage of finite elements is that it is able to predict mechanical properties over complex geometries.

G. Abedrabbo [8] states that “*FEM is well-suited for static analyzes or low dynamics of continuous media*” but require high computational time. Multibody models are in comparison more suited for applications requiring large translational and rotational displacements like gait analyzes and daily life movements. The focus of this thesis is on intervertebral efforts and not on stresses and strains (obtained via FEM), therefore multibody dynamics is more suited.

Multibody dynamic softwares like ADAMS (used in [28] to characterize the mechanical properties of a scoliotic flexible spine) and AnyBody [29] are widely used. Anybody and Opensim (used in [30] to predict joint reactions and muscle forces) are musculoskeletal models for “*biomechanical simulations within the human body or between the human body and an environment*” [29].

Multibody system A multibody system (MBS) is a mechanical system composed of bodies connected by joints. The problem of predicting the motion of the system to which known forces and torques are applied is called *direct dynamics*. Whereas, the *inverse dynamics* focuses on computing the joint forces (torques) inducing the wanted trajectory. The topology of a system, meaning the way the bodies are connected, is described by a tree-like structure (no loop of bodies)(FIGURE 1.3a). When the system is composed of a closed loop of bodies (FIGURE 1.3b), the tree-like structure needs to be restored thanks to a particular procedure, either cutting a body in two or cutting a joint. Parallel robots, four-bar mechanism or even in our case the loop formed by the IMU and the spine (see CHAPTER 2) are closed-loop multibody systems [31].

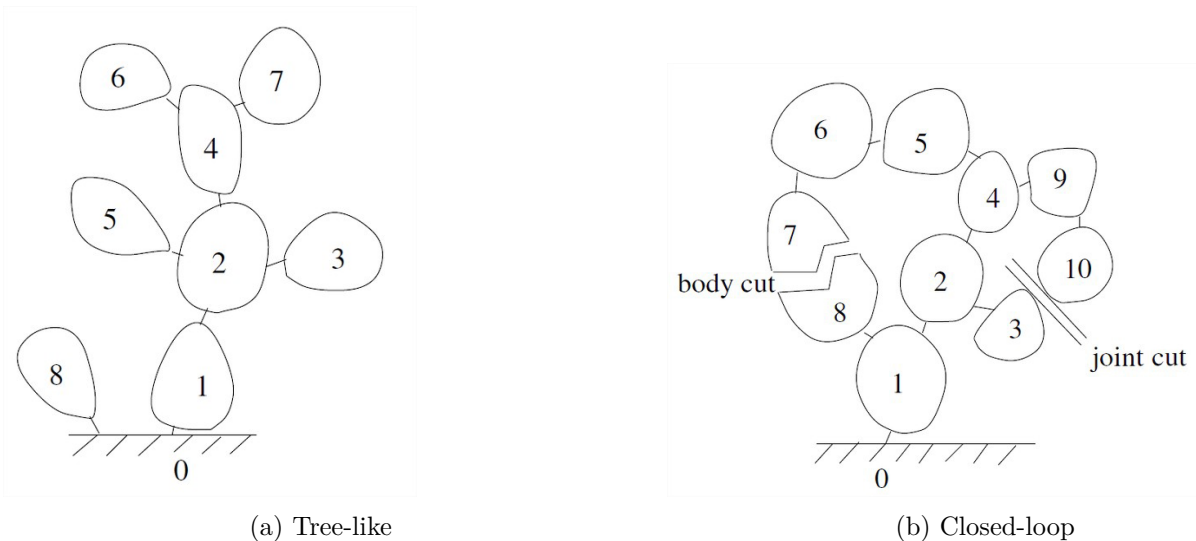


Figure 1.3: Structures of systems [31]

Robotran[32] is a program generating symbolic multibody equations (kinematic, dynamical) of complex physical applications. Using the symbolic approach allows Robotran to simplify remarkably the

mathematical expressions, therefore enhancing the computing speed. Moreover, Robotran is portable, meaning that once the equations related to the mechanical system are generated, there is no need for the program anymore and an interface in Matlab, C or Python, encompassing it is sufficient. Additionally, a visual representation of the system in 3D can be obtained to visualize the simulation results.

Robotran is developed by the Multibody research group of the Center for Research in Mechatronics (CEREM) which is part of institute of Mechanical, Material and Civil (iMMC) engineering institute of the Université catholique de Louvain (UCLouvain).

The Robotran graphical description environnement (MBsysPAD) is mainly composed of bodies, with mass and inertia, and joints. Joints between bodies represent devices such as telescopic arms, ball joints, hinges, etc. Robotran, to avoid describing this large set of joints defined a simple convention. Indeed, each joint only has one degree of freedom and is classified either as *prismatic* (T=translation) or *revolute* (R=rotation) as can be seen in FIGURE 1.4. With these two joints, more complex joints as a cardan joint or a 6 degree of freedom joint can be modeled. Joints are either considered independent, dependent or driven.

- *Independent* : motion of the joint is independent from other joints (# independent variables = # degrees of freedom).
- *Dependent* : motion of the joint is dependent of the other joints (# dependent variables = # constraints).
- *Driven* : motion of the joint is locked or forced to follow a specific law function of the time.

MBsysPAD contains also cuts to separate a body or joint in two and links to model springs or dampers. Those links will be used in CHAPTER 2 when modeling the muscles around the spine.

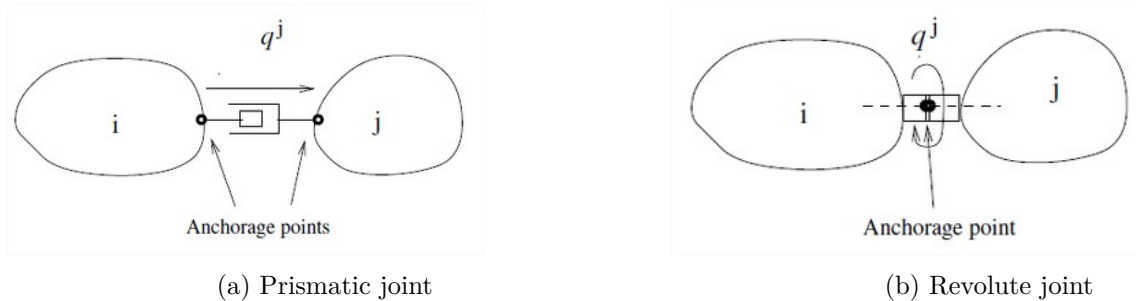


Figure 1.4: Elementary joints [31]

1.3 Application of IMU technology to the field of biomechanics

In order to apply IMU technology to the field of biomechanics, an overview of the IMU needs to be completed first. After that, the main output data of those IMU being the Euler angles, some references need to be defined. Then, the application fields of those IMUs are investigated as well as their calibration method and the problems encountered with the IMUs.

1.3.1 IMU overview

An inertial measurement unit (IMU) is an electronic device consisting of 3-axis sensors providing measurements of the orientation of the platform relative to the Earth (roll ϕ , pitch θ , yaw ψ) and values for the acceleration along the 3 axis. It uses a combination of a gyroscope and accelerometer.

Although useful, the IMUs suffer from a major disadvantage for navigation since accumulated errors appear. Indeed, the acceleration is continuously integrated with respect to time by the guidance system to obtain the speed and position, leading to an accumulation of small errors over time. Therefore, a drift appears, meaning an increasing difference between the actual and the believed position. To diminish this drift, a magnetometer can be used. The IMU is then called an attitude heading reference system (AHRS) platform. With a sensor fusion algorithm, drift coming from the gyroscope data integration can be compensated by the gravity but also Earth's magnetic field.

In the previous master thesis [33], it was decided that the **x-IMU** developed by *x-io Technologies* would be used. Indeed, the x-IMU can easily be connected to the computer and the data of interest can be acquired either via Bluetooth or via a μ SD card. Moreover, UCLouvain employees were already using x-IMUs for years, therefore they had accumulated a certain expertise essential to the completion this master thesis. Therefore, the work was carried out this year with the x-IMU as well wich is an IMU and AHRS platform.

There are 3 main on-board sensors like stated before : an accelerometer, a gyroscope, and a magnetometer.

Accelerometer It provides a measurement of the acceleration along the 3 axis of the x-IMU. The algorithms embedded in the x-IMU use this data to measure the direction of the gravity, therefore providing an absolute reference for the calculation of the pitch and roll angles.

Gyroscope It provides a measurement of the angular velocities around the 3 axis of the x-IMU.

Magnetometer It provides a measurement of the magnetic flux along the 3 axis of the x-IMU. The algorithms embedded in the x-IMU use this data to measure the direction of Earth's magnetic field, therefore providing an absolute reference for the calculation of the yaw angle.

The platform provides six output files (.csv):

- The battery voltage (V) and the temperature ($^{\circ}$ C) (**CalBattAndTherm**).
- The acceleration (g) obtained thanks to an accelerometer, the angular velocity (deg/s) acquired thanks to a gyroscope, and the magnetic field (G) measured thanks to a magnetometer (3-axis values for all sensors) (**CalInertialAndMag**).
- The time (in year, month, day, hour, minute, second) written in the registers of the platform (**DateTime**).
- The Euler angles (roll, pitch, yaw) (degrees) computed by the inner algorithm of the platform (either IMU mode or AHRS mode) (**EulerAngles**).

The IMU mode uses both the gyroscope and the accelerometer sensors. Regarding the AHRS mode, it uses the three embedded sensors on the board, meaning the gyroscope, accelerometer and magnetometer.

- The 4 quaternion elements (other way than Euler angles to measure the orientation of the platform with respect to the earth) (**Quaternion**).
- The rotation matrix from the inertial frame (created when the IMU is tared) to the IMU frame (**RotationMatrix**).

It is important to keep in mind that there is several ways of obtaining the Euler angles of the platform, either with embedded or with external algorithms (see FIGURE 1.5). Indeed, before the measures are

taken, a mode for the embedded algorithm of the x-IMU has to be chosen, IMU or AHRS mode. The IMU mode uses a fusion algorithm combining both the accelerometer and gyroscope data. The AHRS mode uses, in addition to the accelerometer and gyroscope data, the magnetometer data. After the experiment, the x-IMU will create an output file with the Euler angles. But this is not the only way the Euler angles can be retrieved. Indeed, regardless of the mode the x-IMU is in, the platform will create a file containing the raw data of the three sensors of interest. External algorithms can therefore be applied on this raw data and compute the Euler angles as well.

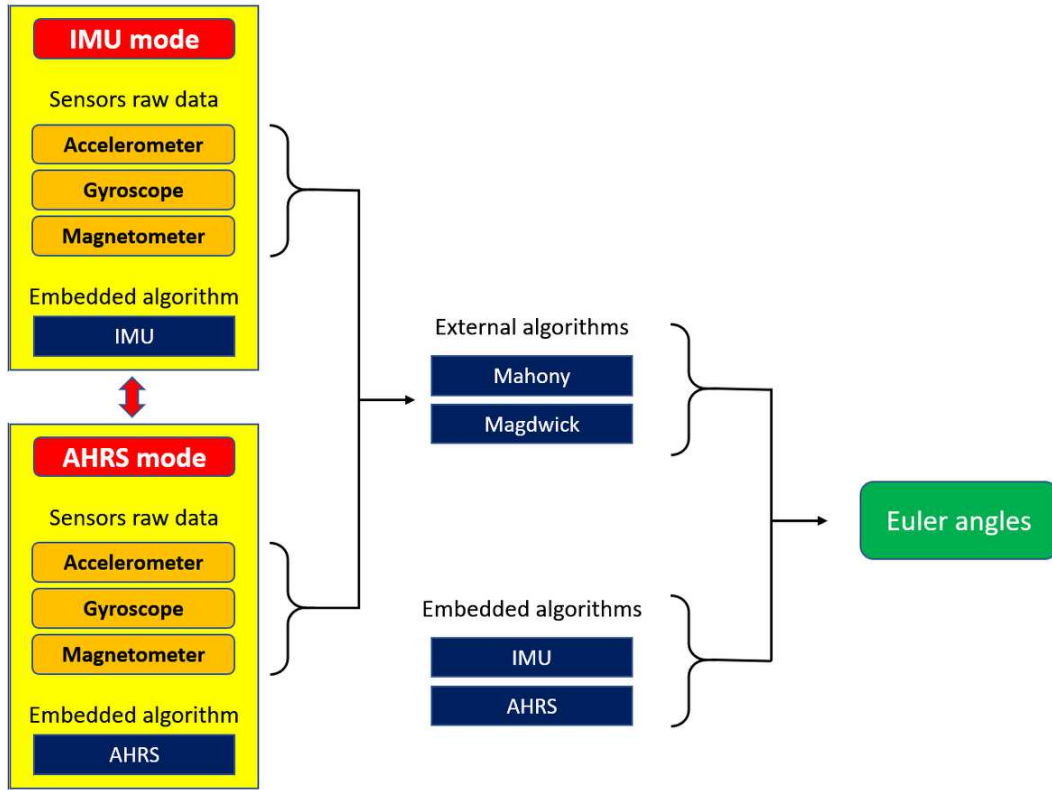


Figure 1.5: Various methods to obtain Euler angles

FIGURE 1.5 shows two of the external algorithms that can be used, the Magdwick and Mahony algorithms.

- **Magdwick** : uses a correction based on the gradient descent optimization algorithm which is an algorithm to find a local minimum of a function, following the negative of the gradient (FIGURE 1.6). The analogy of a bowl and a drop of water could be used. Indeed, the water droplet will fall along the steepest slope of the bowl until it reaches a minima. It also takes into account a possible magnetic distortion due to external magnetic noise (e.g. treadmill).
- **Mahony** : uses a proportional and integral drift correction thanks to a complementary filter mixing the gyroscope, magnetometer, and accelerometer data.

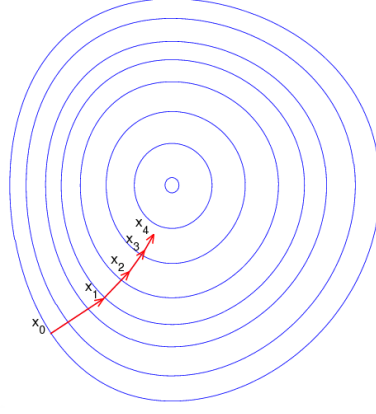


Figure 1.6: Gradient descent algorithm: visualization [34]

SECTION 3.1 depicts the experiment carried out in order to choose the method of data acquisition (mode/algorithm used).

1.3.2 Euler angles

The Euler angles obtained via the IMUs are defined by three successive intrinsic elemental rotations :

1. Rotation around the $\hat{Z}$ axis, by an angle ψ
2. Rotation around the $\hat{Y}$ axis, by an angle θ
3. Rotation around the $\hat{X}$ axis, by an angle ϕ

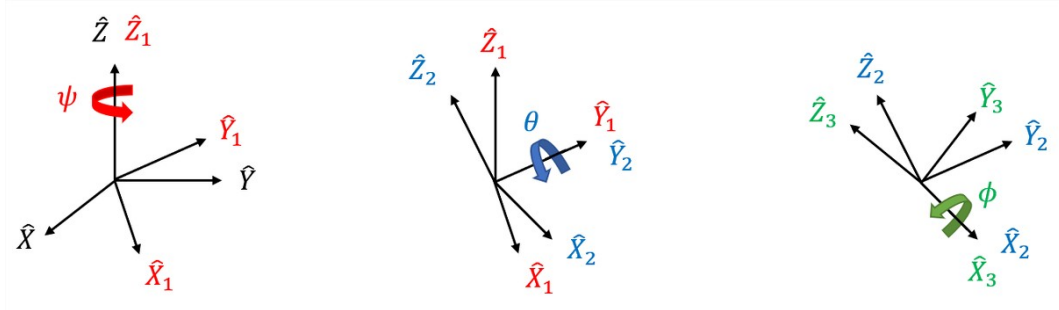


Figure 1.7: Elemental rotations for the ZYX sequence of Euler angles

The resultant rotation matrix can be decomposed as a product of three elemental rotation matrices (sequence ZYX)[35]:

$$\begin{aligned}
 R &= R_z(\psi) R_y(\theta) R_x(\phi) \\
 &= \begin{bmatrix} c_\psi c_\theta & c_\psi s_\theta s_\phi - c_\phi s_\psi & c_\phi c_\psi s_\theta + s_\phi s_\psi \\ c_\theta s_\psi & c_\phi c_\psi + s_\phi s_\theta s_\psi & -c_\psi s_\phi + c_\phi s_\theta s_\psi \\ -s_\theta & c_\theta s_\phi & c_\phi c_\theta \end{bmatrix}
 \end{aligned}$$

1.3.3 Application fields

These last few years, the number of applications of Inertial Measurement Units has increased exponentially. From your phone, to the navigation for space, terrestrial, marine and aircraft field and biomechanical analyses, IMUs are everywhere.

IMUs are widely spread, especially in navigation systems. They are extremely important when Global positioning System (GPS) cannot be used but can be coupled with GPS information for an accurate navigation.

S. Godha [36] states that GPS alone, because it depends on electromagnetic signals, is incapable of providing continuous and reliable positioning. Their subject of research was on land vehicle navigation system (LVNS) technology which aims at accurately locating a vehicle on the road. They evaluated the performance of low cost IMUs integrated with GPS data.

D. Soares dos Santos [37] also merged measurements from a low-cost IMU to a GPS receiver and a digital compass in order to estimate, in real-time, a boat position and heading thanks to a Kalman filter algorithm.

IMUs are also extensively used in aircraft navigation. P. Tomé [38] described a system composed of IMUs and differential GPS. They processed the raw data of the IMUs and the data of GPS from two receivers on the aircraft and more on the ground (on stations). They recovered the attitude, velocity and position of the aircraft and proved that the overall system accuracy was improved compared to the GPS data alone.

The space industry (rovers, interplanetary probes, etc.) also employs IMUs. Those systems need acceleration and angular velocity measurements. S. Clerc [39] presented an analysis on the need for a European Inertial Measurement Unit for space applications.

Navigation is not the only field requiring the use of IMUs. Robotics uses more and more everyday those sensors. In order to control the robot and its behavior, the full motion of the robot needs to be evaluated thanks to both acceleration and angular rate. E. Rocon [40] states that IMU technology can be used jointly with cameras to counteract the accumulated errors of the data integration inside the IMUs.

IMUs can also be used in the biomechanical field for various applications.

Z. Lin [41] tracked body motion thanks to an ultra-miniaturized inertial measurement unit that they developed. They implemented a sensor fusion algorithm based on a Kalman filter to track in real-time the human limb segments.

A. Baghdadi [42] explained a new method for obtaining the hip acceleration and trunk posture in the sagittal plane during gait. They compared two different Kalman filters. They placed the IMUs on the ankle, hip, and middle of the trunk. The IMU situated on the ankle was used as the input for the estimation method and the other IMUs serves as reference for that method.

IMUs can also be used to compute joint angles during gait. T. Seel [43] implemented a method for that using only accelerometer and gyroscope data to be free of magnetic problems. They compared IMU data to an optical 3D motion capture system.

1.3.4 Calibration

The calibration of the x-IMUs was massively explained in the previous thesis [33].

The x-IMUs were compared to an optoelectric motion system. First, a mechanical system rotated the IMUs along their roll, pitch and yaw angles, returning them exactly at their starting point at the end of each experiment. Then random movements were made. The difference between the two sensors were inferior to 5° , the biggest difference occurring in each experiment in the yaw angle.

For a 6 km/h walk, the roll, pitch and yaw values of the IMU situated on the pelvis only had differences of 2° between the IMU and the optoelectric motion system.

1.3.5 Problems encountered and their resolution

The previous thesis [33] states three main problems occurring when using the x-IMUs :

- **Magnetic perturbation** : they analyzed the magnetometer perturbations occurring when a treadmill was powered on and increased in power. It was observed that there was a significant change in the three components values of the magnetometer, therefore influencing the Euler angles. It was stated that the walking experiment should be realized in an environment where the external magnetic fields were as low as possible in order to get results as precise as possible.
- **Imprecision at high speed** : random "slow" and "fast" movements (the exact values were not stated) were realized and compared to optoelectric motion system values. In the slow movement, the biggest errors stayed under 5° . The quantification of the errors was not given for the fast movement but, analyzing the figures of the results obtained, a high number of disparities could be observed. It is therefore explained in their master thesis that the IMUs should be used at low speed.
- **Discontinuity of the Euler angles** : they state that the analysis of movements is more complicated since an angle of 180° on the IMU could correspond in reality to an angle of 360° or even 720° , the number of turns not being accounted for in the IMU. Indeed, Robotran cannot work with discontinuous angles. They developed a function in order to count the number of turns the IMU performed.

1.4 Discussion of the state-of-the-art

In order to formulate a research question on the large subject of the thesis, the state-of-the-art had to be analyzed.

To correctly assess the dynamics of the lumbar spine, the kinetics of it had to be measured, meaning the spine motion. It was seen that the experiments were nowadays realized with systems allowing a continuous data acquisition in order to fully assess the full range of motion of the spine in movement. A lot of experiments are conducted with optoelectric system motion capture coupled with reflective markers affixed on the skin above the vertebrae. This method is expensive and a laboratory is needed in order to perform the experiments. With Inertial Measurement Units, the experiments are freed from laboratories constraints because they are portable. Indeed, if wanted, the experiments could be performed on the top of a mountain for example. Moreover, IMUs are more affordable than optoelectric systems and they are quite precise (5°) as stated in [33] where they compared IMUs to the cameras. Daily life movements can be tested in this thesis as in G. Abedrabbo's PhD thesis [8] but the most important difference is that the data acquisition protocol would be simpler here since his results are obtained via the optoelectric system.

The previous thesis [33], which is the base for mine, states that there is imprecision in the IMU data at high speed, but the daily life movements are mainly performed at low speed, being quasi-static, so this is not something to be worried about in this master thesis.

The biggest gap found in the literature is that there is no researches on the computation of intervertebral efforts of the lumbar spine thanks to data acquisition with inertial measurement units.

A new article from May 2019 [44] assessed the spinal posture during gait thanks to three inertial sensors positioned on the pelvis and thorax. This proves that evaluating the kinematics and dynamics of the spine is at the cutting edge of scientific research.

The following questions will be answered all along the length of this master thesis :

- The subject of patient-specific models is touched upon with respect to the spine geometry. But a question we may ask ourselves is if an inertia change of the body segments induces changes in the intervertebral efforts of the lumbar spine.

- A magnetic perturbation was noticed in the previous thesis [33] with a treadmill. But what about specific magnetic devices sending pulse of magnetic field in order to synchronize IMU data ... Would it have an impact on the Euler angles ?
- The discontinuity of Euler angles in the IMU results is not a real problem since a simple function can be written in order to take into account the number of turns and remove this small inconvenience. This will be shown later.
- The IMUs seem to be less precise for their yaw value. This will be verified ... But is that really a problem for the application realized in this master thesis ?
- The mode and algorithm of the IMU was not stated in the previous master thesis [33], therefore it was not tested if a mode or algorithm had better results than another. An experiment will be carried out in order to choose the best mode and algorithm for the use of the IMUs.
- One could ask himself how many IMUs are needed to fully assess the intervertebral forces of the lumbar spine.

1.5 Objectives

Nowadays, it becomes important to provide surgeons specialized in spinal operations a model of the spine (kinematics and dynamics) in order to help them decide which surgery technique is the most suitable for each patient. It could also be useful in sports in order to follow high level athletes and prevent them to injure themselves. Moreover, manufacturers of protective gears could also be interested in such models. Many different fields could use models in order to compute the different efforts in the lumbar spine. In order to obtain the data inputs of those models, it is also important to simplify the data acquisition protocol. To this end, the goal of this master thesis is threefold:

- Discuss the state-of-the-art regarding the following two subjects : the biomechanics of the spine and its multibody modeling, and the application of IMU technology to the field of biomechanics.
- Apply the IMU technology to a multibody spine model based on the model of G. Abedrabbo [8] to find the kinematics and dynamics of the spine for various simple movements of the daily life and during gait movement. To this end, program a full computer implementation in C for the model of the spine.
- Offer a perspective/an opinion on IMU technology applied to the modeling of the spine.

In CHAPTER 1, a state-of-the-art on the biomechanics of the spine, its modeling, and the application of IMU technology to the field of biomechanics was carried out in order to understand where the research stands nowadays on those subjects. The goal of this literature review was to find gaps in those researches and formulate research questions based on those gaps.

CHAPTER 2 will describe the modeling process of the multibody model of the spine and the integrations of the IMUs in that model. To this end, a number of steps will be conducted in order to find the intervertebral efforts in the lumbar region and the computer implementation of those steps will be explained.

CHAPTER 3 will focus on the analysis of the results obtained via the different experiments conducted within the framework of this master thesis.

CHAPTER 4 will conclude this report, offer perspectives on the work achieved for this master thesis and propose future researches on this subject.

This chapter explains the modeling process of the complete model composed of the spine and the IMUs. First, a small introduction to the subject will be given, stating the main hypotheses made in order to simplify the model. Then, in SECTION 2.1, the modeling of the spine will be explained, along with the computation of anthropometric data like the mass and inertia of the bodies composing the spine, and their inclusion in the model. SECTION 2.2 will explain the integration of the IMUs in the spine model while SECTION 2.3 will state the steps realized to this end, and SECTION 2.4 will explain the computer implementation of those steps in the Matlab and C languages.

The dynamics of the spine, especially the intervertebral disks efforts, are an important source of information in surgical planning. Indeed, it is important to quantify those forces in order to better understand for example the excessive energy cost of motion for scoliotic patients. But to obtain the dynamics of the spine, the kinematics of it, meaning its motion, has to be acquired. One of the ways to obtain the kinematics of the spine is to use inertial measurement units (SECTION 1.3). To analyze and transform the data coming out of the IMUs into usable data for the spine, a model has to be built. The spine and the IMUs are modeled in Robotran (SECTION 1.2.4).

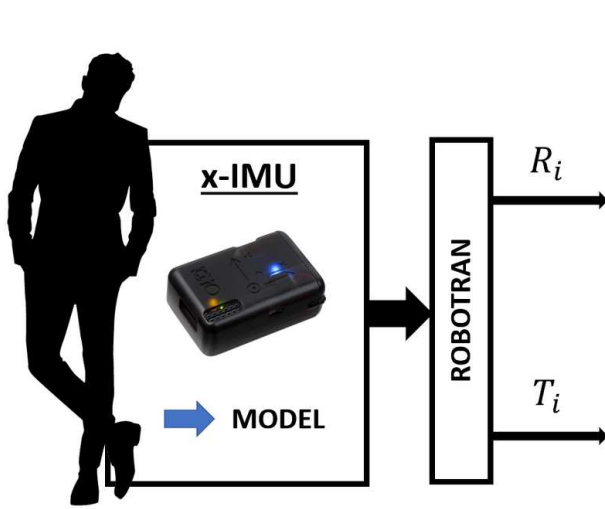


Figure 2.1: Spine kinematics [45][46]

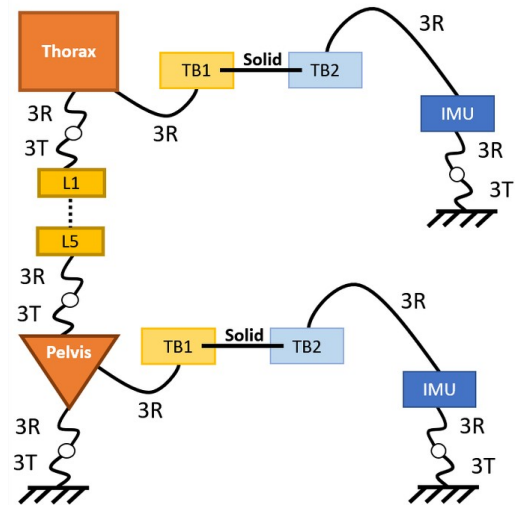


Figure 2.2: Multibody model of the spine and IMUs (R=rotations, T=translations)

Thanks to this model (FIGURE 2.2) and the fact that the geometry of the spine is known at time zero, it is possible to retrieve the angles and translations between the vertebrae as a function of the time (FIGURE 2.1). It is also possible to retrieve the rotations and translations of the pelvis, but those translations (and the yaw angle) are called, in the multibody language, non-sensitive variables. It means that they can be ignored. You do not want to know where you are in the room or which wall you are facing, but you want to know the kinematics of the spine itself. Non-sensitive variables are often called parking variables because, for a car, they are just absolute positions and orientation. It does not matter for the kinematics of the car if it is parked facing north or west or its position in the parking lot.

One could ask himself, why achieve data acquisition using IMUs and not optokinetic sensors, which are reflective markers stuck on the skin of the patient in addition to cameras recording the 3D coordinates of the markers. The answer is simple. To use the latter, you have to go to a special laboratory and use a treadmill. For the former, you can be at home or outdoors for the data acquisition. It is therefore less restraining for the patients. Indeed, for a gait analysis, 3 steps in your room are sufficient.

Another question that may arise is the number of IMUs used. The first solution that comes into our minds is that the more IMUs we have, the better. But putting for example 7 IMUs on the spine of a patient is impossible due to the size of the sensor (57 x 38 x 21 mm). Even with the optokinetic sensors, attaching a reflective marker to each vertebra was impossible[33]. Based on the upcoming hypotheses and the experiment in SECTION 3.5, it was speculated that 3 IMUs were sufficient for computing the kinematics of the spine: one on the pelvis (S1) and two on the thoracic cage (one the lowest possible (Th12) and the other the highest possible ($\approx$ Th6)) (FIGURE 3.21). But with this low number of IMUs, the rest of the spine had to be reconstructed. Two hypotheses had to be made:

- The thoracic cage is assumed to be rigid. This hypothesis will be tested during the experiment in SECTION 3.5. Indeed, 3 IMUs will be attached to the test subject and only 2 IMUs will be modeled, one for the pelvis, and one for the thoracic cage. The data coming from the two real IMUs fastened on the thorax of the test subject will be the input of the virtual IMU affixed on the modeled thorax, and the simulations will be compared.
- Each intervertebral disk in the lumbar region is reduced to a single massless point and can be represented by a series of 3 rotational springs representing 3 degrees of freedom in rotation between each vertebra with a stiffness given by TABLE 2.1 and a damping 10 times smaller. The degrees of freedom related to the 3-dimensional translations of the vertebrae with respect to each other are assumed to be fixed. Indeed, [47] gives values for those translational stiffnesses as high as $2420 \left[\frac{kN}{m}\right]$. It can be said that, since the translational stiffnesses are high, the deformations of the intervertebral disks are minimal in translation. It is assumed in this master thesis that, the translation between vertebrae being fixed and constant, there is only 3 degrees of freedom between each vertebra.

In the study conducted in [47], they only had the stiffnesses for the L2-L3 and L4-L5 disks. Therefore, the value of the L2-L3 disk stiffness was applied to Th12-L2, too. Then, the L4-L5 disk stiffness was applied to L5-S1. For the L3-L4 disk, the mean of both values was chosen.

Motion [N.m/rad]	Th12-L3	L3-L4	L4-S1
Roll	236	306.5	377
Pitch	287	431	575
Yaw	734	783	832

Table 2.1: Stiffnesses of the lumbar intervertebral disks [47]

TABLE 2.1 presents the coupled stiffness parameters. Indeed, there is a coupling between intervertebral motions. As explained in [47], with forward flexion of the joint, it tends to produce translation as well.

A first idea was to take uncoupled stiffness parameters since the model would only have 3 degrees of freedom in rotation between each vertebra. But these values being too small ($R_{\phi,\theta,\psi} \approx 70, 95, 550 \left[\frac{Nm}{rad}\right]$) [47], when a force representing the muscles and computed thanks to the IMUs

was applied on the spine model, it would collapse, giving wrong results. Indeed, these uncoupled parameters do not represent the reality of the spine. Moreover, higher stiffness values were needed because the spine is in reality surrounded by a numerous amount of things, like the skin or the muscles, giving the body a higher stiffness in general.

2.1 Spine model

In order for the spine model to match with precision to the test subject, one should apply the stereography method to the person. But this master thesis is dealing with healthy subjects and obtaining x-rays for a healthy patient is not an easy thing to justify. Moreover, since this is a master thesis and not a PhD thesis, there is no possibility to perform x-rays on the test subject.

This model is based on the model created by G. Abedrabbo [8]. The geometry of the spine was obtained thanks to the stereoradiography method applied to a scoliosis patient. Since this master thesis does not focus on scoliosis and that the subject for the experiments will not be a scoliotic patient, the spine was straightened (roll and yaw put initially to zero). All the translations in this model (FIGURE 2.3) are fixed.

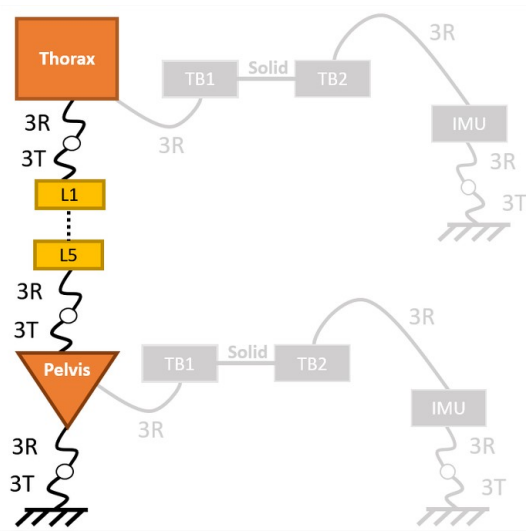


Figure 2.3: Spine model

There are three main parts in this model :

- **The thorax** : the 12 vertebrae are fused together, being represented in FIGURE 2.3 by a single body.
- **The lumbar spine** : the 5 vertebrae are separated by 3 degrees of freedom in rotation since the translations are fixed.
- **The pelvis** : it is represented by a single body with 3 degrees of freedom with respect to the inertial frame of the model. The translations are also fixed.

All the rotations between the thorax and the pelvis are part of the data of interest in this master thesis.

2.1.1 Changes with respect to the given model

G. Abedrabbo's model was used only for the geometry of the spine, meaning the prismatic and revolute joints values of the system. The translations were imposed thanks to vector coordinates (dpt points in Robotran) and not prismatic joints.

Several modifications were implemented for the modeling of the spine :

- 3 prismatic joints were added in the lumbar spine between each vertebra to be able to recover shear and normal efforts in the intervertebral disks. The translations values were still imposed via dpt points.
- The mass and inertia of the bodies making up the spine were calculated and inserted in the model (see SECTION 2.1.2).
- A body representing the thorax, head, and arms total inertia and mass was added.

2.1.2 Calculation of anthropometric data

Anthropometry is the study of sizes and proportions of the human body. This section will focus on calculating the mass and inertia of the different body segments.

Mass

The masses can be found in literature [48] as a percentage of the total mass of the human body.

Segment	%	Mass [kg]
Head	6.68	3.7408
Upper part of trunk (UPT)	15.45	8.6520
Middle part of trunk (MPT)	14.65	8.2040
Lower part of trunk (LPT)	12.47	6.9832
Upper arm	2.55	1.4280
Forearm	1.38	0.7728

Table 2.2: Percentage of the different segments of the female human body with respect to its total mass and their mass for a 56 kg female human body. [48]

The mass of the additional body (see SECTION 2.1.1) is the sum of the masses of the head, the upper part of the trunk, the upper arms and forearms. The mass of each vertebra is defined in this master thesis to be the MPT mass divided by the number of lumbar vertebrae. The mass of the pelvis is considered to be the one of the LPT.

Bodies of interest	Mass [kg]
Additional body	16.7944
Lumbar vertebra	1.6408
Pelvis	6.9832

Table 2.3: Mass of the bodies of interest for a 56 kg female human body

Inertia

This section will focus on calculating the inertia of the additional body, each lumbar vertebra and the pelvis.

There is 2 ways of calculating the inertia :

- For the upper (additional) body and the pelvis, the inertia is calculated thanks to experimental data from litterature (expressed as radius of gyration).
- For the lumbar spine, since sufficient data is not available, it was chosen to model the vertebrae as cylinders, therefore using a simple formula to compute the inertia.

In order to understand the tables found in the literature, a definition of the planes and axis of the human body needs to be given. In this report, the inertial frame of the model is oriented in such a way that it overlaps with the anatomical planes of the human body. In FIGURE 2.4, the planes formed by $\{\hat{I}_1, \hat{I}_2\}$, $\{\hat{I}_2, \hat{I}_3\}$, and $\{\hat{I}_1, \hat{I}_3\}$ are respectively named the transverse, frontal and sagittal planes.

The roll, pitch, and yaw angle are defined with respect to the sagittal axis ($\hat{I}_1$), the transverse axis ($\hat{I}_2$), and the longitudinal axis ($\hat{I}_3$) respectively.

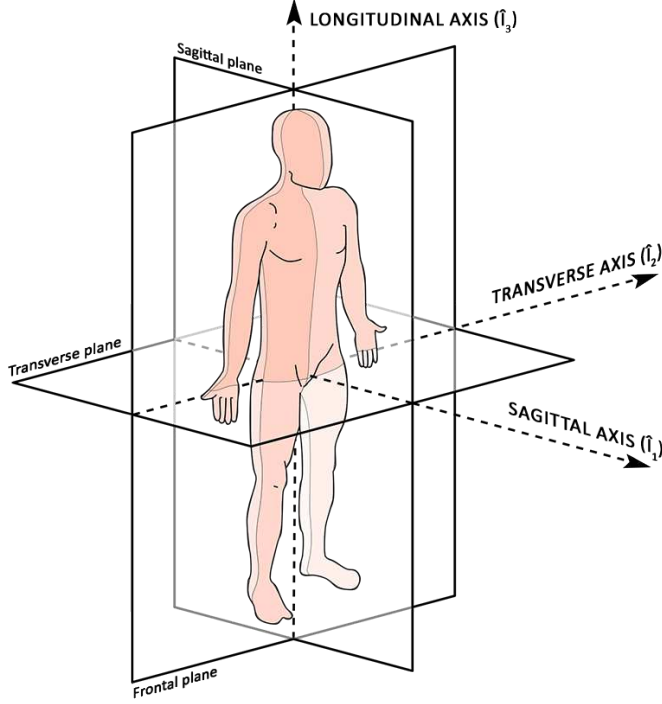


Figure 2.4: Anatomical planes and axes (adapted from [49])

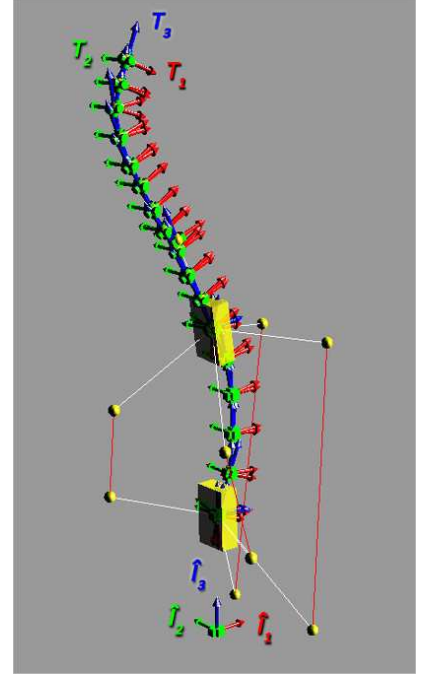


Figure 2.5: Frames in the visualization tool of Robotran

Radius of gyration According to the dictionary [50], the radius of gyration is defined as “the distance from an axis at which the mass of a body may be assumed to be concentrated and at which the moment of inertia will be equal to the moment of inertia of the actual mass about the axis”. It is equal to the square root of the quotient of the moment of inertia and the mass.

Therefore, to obtain the inertia :

$$I_{axis} = m \cdot r_{g\ axis}^2$$

TABLE 2.4 presents the longitudinal length of different segments of the female human body. It also shows the center of mass (CM) of those segments as a percentage of their length. The segment CM positions are referenced either to proximal or cranial endpoints (see FIGURES 2.6 and 2.7). Both segments CM position and radius of gyration (r) are relative to the segment length.

Segment	Long. length [mm]	Long. CM position [%]	Sagittal r [%]	Transverse r [%]	Long. r [%]
Head	200.2	58.94	33.0	35.9	31.8
UPT	142.5	20.77	74.6	50.2	71.8
LPT	181.5	49.20	43.3	40.2	44.4
Upper arm	275.1	57.54	27.8	26.0	14.8
Forearm	264.3	45.59	26.1	25.7	9.4

Table 2.4: Longitudinal length of different segments of the female human body, center of mass position, and sagittal, transverse and longitudinal radii of gyration. [48]

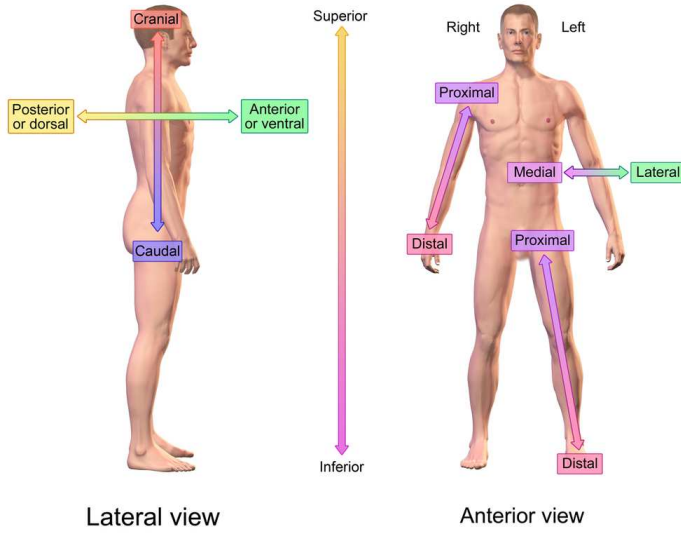


Figure 2.6: Anatomy terms of location [51]

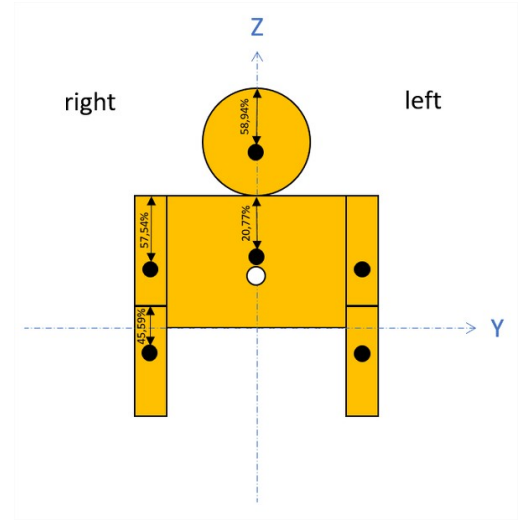


Figure 2.7: Location of the CM for the additional body

The values of the table cannot be used directly to compute the inertia matrix of the additional body representing the head, the thorax and the arms. Indeed, the first step to obtain the joint inertia is to find the center of mass of those combined bodies. To obtain it, a reference frame is needed ¹ (FIGURE 2.7) and coordinates along the 3 axis for the center of mass of each body needs to be computed ². Once this step is finished, the joint center of mass can be computed thanks to the formula :

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}_{CM} = \frac{1}{m_T} \begin{bmatrix} \sum_{i=1}^n (m_i \cdot x_i) \\ \sum_{i=1}^n (m_i \cdot y_i) \\ \sum_{i=1}^n (m_i \cdot z_i) \end{bmatrix}$$

with n the number of bodies, m_T the total mass of the system, and (x_i, y_i, z_i) the 3D coordinates of the center of mass of body i .

¹The origin of this reference frame (0,0,0) is represented by the T12 vertebra.

²The longitudinal values are given in TABLE 2.4 and a distance of 18 cm was measured on the test subject between the shoulder and the sternum.

This center of mass will serve as the reference point for the inertia. The value (0,0,0.0822) was computed for the joint center of mass. Indeed, the x and y values were expected to be zero since the x_i values were considered to be null and the system is symmetrical.

The inertia matrix of each part of the system can now be expressed with respect to one point only, the joint center of mass. To this end, the theorem of Huygens-Steiner [52] can be used. This theorem states that the inertia of a body with respect to a point different from its center of mass is equal to the sum of the inertia of the body with respect to its center of mass, and the inertia due to the energy of transport of the body. Indeed, it is more efficient from an energetic point of view to rotate a body around its center of mass (G) than around any other point (P).

$$I_P = I_G + m \cdot \begin{bmatrix} b^2 + c^2 & -ab & -ac \\ -ab & a^2 + c^2 & -bc \\ -ac & -bc & a^2 + b^2 \end{bmatrix} \quad \text{with} \quad \overrightarrow{GP} = \begin{bmatrix} a \\ b \\ c \end{bmatrix} \quad \text{and} \quad I_G = m \cdot \begin{bmatrix} r_g^2 \hat{X} & 0 & 0 \\ 0 & r_g^2 \hat{Y} & 0 \\ 0 & 0 & r_g^2 \hat{Z} \end{bmatrix}$$

To obtain the final matrix of inertia, the I_P values for each body of the system were added together. A final step was achieved by taking only the values on the principal diagonal of the matrix, meaning $I_{G,11}$, $I_{G,22}$ and $I_{G,33}$ since the other values were negligible. The value (0.5661, 0.3706, 0.2540) [$kg \ m^2$] was obtained for the inertia with respect to the joint center of mass of the entire system. The matlab file can be found in APPENDIX B.2.

For the pelvis, since it was only one body, the calculation was easier : $I_{axis} = m \cdot r_{axis}^2$

Cylinder For a solid cylinder of radius r , height h and mass m [53] :

$$I_x = I_y = \frac{1}{12} m (3r^2 + h^2) \quad I_z = \frac{1}{2} m r^2$$

With a radius of 10 cm ($\approx$ waist) and a height of 3 cm, the inertia is equal to (0.004225, 0.004225, 0.008204) [$kg \ m^2$].

Bodies of interest	$I_{\hat{X}} [kg \ m^2]$	$I_{\hat{Y}} [kg \ m^2]$	$I_{\hat{Z}} [kg \ m^2]$
Additional body	0.566100	0.370600	0.254000
Lumbar vertebra	0.004225	0.004225	0.008204
Pelvis	0.043130	0.037176	0.045350

Table 2.5: Inertia of the bodies of interest (with respect to their CM) for a 56 kg female human body

Therefore, the additional body is modeled thanks to a body of mass 16.7944 [kg] and inertia (0.5661, 0.3706, 0.2540) [$kg \ m^2$] and attached to the T12 vertebra thanks to a dpt point (=vector) of value (0, 0, 0.0822).

2.1.3 Model improvements

Once the anthropometric data was embedded in the model, problems occurred in the simulations. Either the model was subject to vibrations or the spine would collapse in itself, passing by a singular position. Moreover, problems would occur more easily in the simulation where the spine was driven by the muscles. Indeed, the muscles tend to pull on the spine in a different way than the simulation with the IMUs where it guided the rotation of the pelvis and the upper body.

Let us keep in mind that the hypothesis made for the tables and the cylinders were not perfect. In fact, the tables consider the body segments as rigid blocs, but in reality there is a certain flexibility to the bodies. The spine is not the only part composing the human body. Numerous muscles, skin, organs, etc. enclose the spine, therefore changing its inertia. The computation of the inertia for the lumbar spine is also based on big assumptions since its calculation is based on rigid cylinders. The entirety of the segment mass is supposed to be interdependent of the vertebrae but this is not the case (e.g. internal organs). These inaccuracies in the calculation of the inertia need to be taken into account if, and only if, the intervertebral efforts are affected. An in-depth sensitivity analysis needs to be conducted but this will not be the focus of this thesis.

To reduce vibrations, the dampening value of the lumbar region was augmented from $K/10$ to $K/8$, the stiffness values being those of TABLE 2.1. Moreover, the inertia of the 3 main parts of the body (upper part, lumbar region and pelvis) was increased until no more vibration or collapsing of the spine could be observed for all the different movements.

The study of the effect of a change in inertia in the upper body, lumbar spine or pelvis, on the intervertebral disks efforts has been carried out in SECTION 3.2. Thanks to this experiment, TABLE 2.6 could be completed.

Bodies of interest	$I_{\hat{X}} [kg m^2]$	$I_{\hat{Y}} [kg m^2]$	$I_{\hat{Z}} [kg m^2]$
Additional body	5.66100	3.70600	2.54000
Lumbar vertebra	0.21120	0.21120	0.41020
Pelvis	0.43130	0.37176	0.45350

Table 2.6: Inertia of the bodies of interest (with respect to their CM) after the change: additional body (x10), lumbar vertebra (x50) and pelvis (x10)

2.2 Integration of the IMUs in the multibody spine model

The first step was to create an IMU model in order to include it later in the spine model.

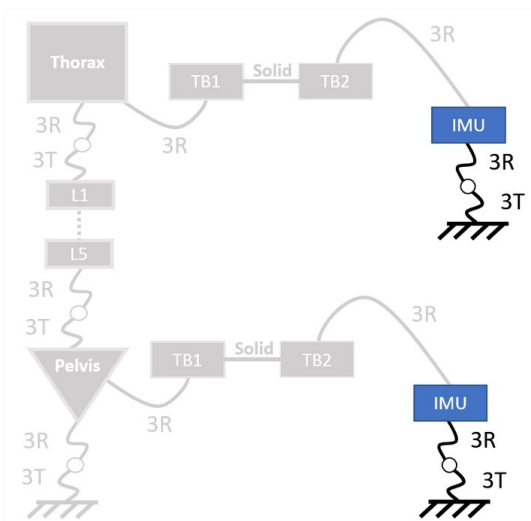


Figure 2.8: IMU model

The model on FIGURE 2.8 represents the IMUs in the inertial frame of the model. The idea is to control a virtual IMU via the three Euler angles recovered from the real IMU: the roll, the pitch, and the yaw. The IMUs have 3 more degrees of freedom in translation to be able to be fixed on the spine.

Different models before this one were built, first in Matlab, then in C, to understand how to translate the IMU data into a simulation and recoverable results (APPENDIX A).

In order to obtain a usable model, this IMU model has to be integrated to the spine model.

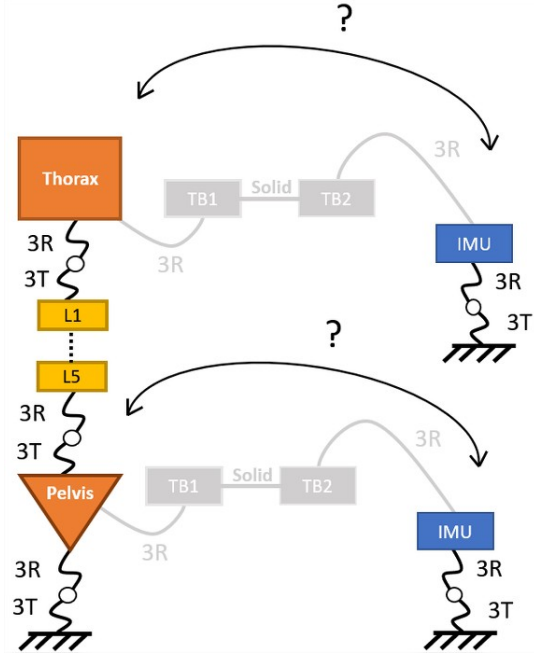


Figure 2.9: Missing link between the spine model and the IMU model

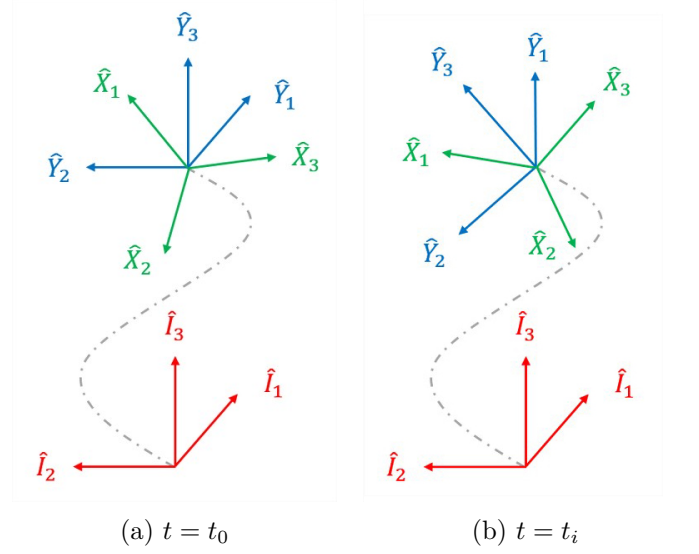


Figure 2.10: Schematic of the different frames used in this model

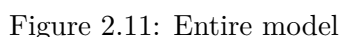
At time $t = 0 = t_0$, the spine has a specific geometry composed of rotations and translations between each vertebra. The IMUs also have roll, pitch, and yaw values different from zero at time t_0 because they have already been fixed on the test subject when the recording of data starts. The goal of this final step is to link the IMU model with the spine model, taking into account the rotations of the spine and the IMUs at time zero.

At the starting time, the rotations of the IMUs with respect to the inertial frame are already different from zero. But these initial rotations have no interest for us because we are interested in the rotations after t_0 , meaning the kinematics of the spine during its motion. Therefore, an additional body aligned with the inertial frame $\{\hat{I}\}$ at time t_0 has been added. We will call it *transition body* (TB2 in FIGURE 2.9). It is represented by the frame $\{\hat{Y}\}$ in FIGURE 2.10. $\{\hat{X}\}$ represents the frame interdependent of the IMU. A transition body aligned with the inertial frame has been added for the spine model, too (TB1 in FIGURE 2.9). The 3 rotations between $\{\hat{X}\}$ and $\{\hat{Y}\}$ allow us to have, at time $t = 0$, $\{\hat{Y}\} = \{\hat{I}\}$.

We now have two transition bodies, each of them fixed to either an IMU or the spine and aligned with the inertial frame at t_0 . The linking of the motion of the IMU with the spine is achieved thanks to a solid cut as seen in FIGURE 2.9. A solid cut adds 6 constraints to the system (3 rotations and 3 translations), fusing the two transition bodies together. When one of the transition bodies moves during the simulation, the other follows.

2.3 Simulation steps

In order to compute the kinematics of the spine thanks to data coming from the IMUs, a few steps have to be implemented. This section will explain the steps to obtain, first, the kinematics of the spine, then, the intervertebral forces.



First of all, it is not an easy task to calculate the orientation of the vertebrae in the inertial frame since there are many vertebrae, therefore rotations, between the base and the vertebra in question. One cannot simply add up all the rotations separately per axis. The easiest thing to do is to let Robotran calculate that automatically via the three rotational joints labeled (3) in FIGURE 2.11.

State of the joints



- ① driven (=translations via dpt points in .mbs)
- ② driven (=initial rotations of the vertebrae with respect to the previous one)
- ③ dependent $\rightarrow$ what we want to find
- ④ driven ($R_{1,2,3}=0$)
- ⑤ driven ($R_{1,2,3}=0$)
- ⑥ dependent (on the position of the vertebrae in the space)

Verification

The values expected and obtained for the pelvis ③ are 0 [rad] for all Euler angles since ② values were also equal to 0 [rad], and initial rotation of the pelvis was indeed not wanted. For the thorax, since all roll and yaw angles of the spine were put to zero to straighten it, roll and yaw angles for ③ were expected to be 0 [rad] and a value different from zero was expected for the pitch angle.

Euler angles [rad]	Value expected	Value obtained
	<u>Thorax</u>	
Roll	0.0	0.0
Pitch	$\neq 0.0$	0.432456
Yaw	0.0	0.0

Table 2.7: Values expected and obtained for the angles between TB1 and the thorax

2.3.2 Step 2: calculate the rotations of the transition body with respect to the IMU

The second step is now to calculate the rotations the transition body has with respect to the IMU. This calculation will, again, be done by Robotran automatically in a one-shot simulation as in step 1.

State of the joints

To calculate ④, we need the joints to be in a certain state. The joints ①, ②, and ⑥ are still the same as in SECTION 2.3.1. In the previous step, certain values were obtained for ③. The inverse of the angles of the vertebrae with respect to the inertial frame are fixed to those values for this step, because there is a solid link between the two transition bodies. Indeed, the first transition body has to be already aligned to the inertial frame for the second to be aligned too since they are fused together. ⑤ is fixed at the values of the Euler angles of the IMU at time zero since we want to calculate the angles in ⑤ leading to an horizontal IMU at time zero.



- ① driven (=translations via dpt points in .mbs)
- ② driven (=initial rotations of the vertebrae with respect to the previous one)
- ③ driven (=fixed to the values calculated in SECTION 2.3.1)
- ④ dependent → what we want to find
- ⑤ driven ($R_{1,2,3} = cst = f(t_0)$)
- ⑥ dependent (on the position of the vertebrae in the space)

Verification

The verification of this step will be done for one IMU only in this report, but it is the same modus operandi for all of them.

Euler angles [rad]	IMU value ⑤	TB2-IMU angle ④
Roll	-0.294380	-0.231310
Pitch	1.538100	-1.538946
Yaw	-0.301830	0.240826

Table 2.8: Values obtained for the IMU Euler angles and the angles between TB2 and the IMU

The Euler angles given by the IMU are following the ZYX sequence. Therefore, the rotation matrix from SECTION 1.3.2 can be used in this case to compute the rotation of an arbitrary vector. Let us take (1,2,3) as the initial vector. After passing through the two rotations of TABLE 2.8, it should come back to its initial value.

$$\begin{bmatrix} x_f \\ y_f \\ z_f \end{bmatrix} = R_4 \cdot R_5 \cdot \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$

Vector	Value expected	Value obtained
x_f	1	0.9993
y_f	2	1.9999
z_f	3	3.0003

Table 2.9: Values expected and obtained for the final vector

The small difference can be explained by the fact that the numbers displayed by the console used for the C program rounds numbers to 6 digits after comma. The values inside the program are therefore more precise than those given in TABLE 2.8. But this small verification proved that the values obtained are the same as those expected. This step can therefore be validated.

An additional validation has been performed. The values of ⑥, which are the translations of the IMU, should remain the same during both step 1 and step 2, since it is fixated on the same spine. The values matched perfectly.

2.3.3 Step 3: obtain the spine kinematics

The third step is to obtain the kinematics of the spine by driving the spine model by the values obtained from the real IMUs ($f(t)$). The acceleration of the pelvis was added in this step.

State of the joints

To calculate ②, we need the joints to be in a certain state. Joints ① and ⑥ are still the same as in the two previous subsections. Regarding joints ③ and ④, they are fixed to the values (=constants) obtained previously. The three rotations of the IMU are now driven by the values obtained from the real IMU ($f(t)$).



- ① driven (=translations via dpt points in .mbs)
- ② dependent → what we want to find
- ③ driven (=fixed to the values calculated in SECTION 2.3.1)
- ④ driven (=fixed to the values calculated in SECTION 2.3.2)
- ⑤ driven ($R_{1,2,3} = f(t)$)
- ⑥ dependent (on the position of the vertebrae in the space)

All the rotations of the vertebrae between the one linked to the IMU and the pelvis (2') are independent, (2) being the only dependent rotations. An explanation for that is that we add 6 constraints by closing the loop (solid cut) so we only need to choose 6 dependent joints among the joints in the loop. (2) and (6) were chosen.

2.3.4 Step 4: obtain the spine dynamics

The fourth step is conducted in order is to obtain the intervertebral efforts of the spine. To this end, this step will integrate the muscular forces to the model. The dynamics cannot be retrieved via the previous step because an important part of the model is missing, the muscles, and the forces would not be representative of the reality. Indeed, the spine moves thanks to the force applied by the muscles on it. SECTION 3.3 will dive into the differences between the forces retrieved via the third step and those retrieved when the spine is driven by muscles.

To obtain the dynamics of the spine, the thorax was driven by muscles and the pelvis by the values obtained from the real IMU attached to the pelvis ($f(t)$) as in the previous step. Indeed, the muscles driving the thorax are attached to the pelvis so the anchoring body has to follow the kinematics it previously followed. The only way to obtain that is to drive the pelvis in the model with the IMU attached on the pelvis of the person.

One could ask himself why the legs were not modeled (FIGURE 2.12). This legged model would be a perfect example of the direct dynamics approach because thanks to the leg forces as inputs of the model, the pelvis motion could be recovered. The inverse dynamics base itself on the kinematics of the pelvis, whatever the source of the movement is. Indeed, the pelvis kinematics is a constrained motion to which constrained forces and torques are associated. The causes leading to the movement are not interesting in our case here. If a person has an asymmetrical way of walking due to an injury to one leg, this will be reflected in the kinematics of the pelvis. Thanks to the IMU on the pelvis, the kinematics and acceleration of it are known which are indispensable to compute the forces in the spine. Indeed, the acceleration is important too, because for example when a person drives a car over a pothole, the shock is more important to the dynamics of the spine than the movement of it. The legs and external forces are therefore replaced by the kinematics of the pelvis and its acceleration. One could think of the Thévenin or Norton equivalent in electricity, meaning replacing a complex electrical network by something equivalent but simpler.

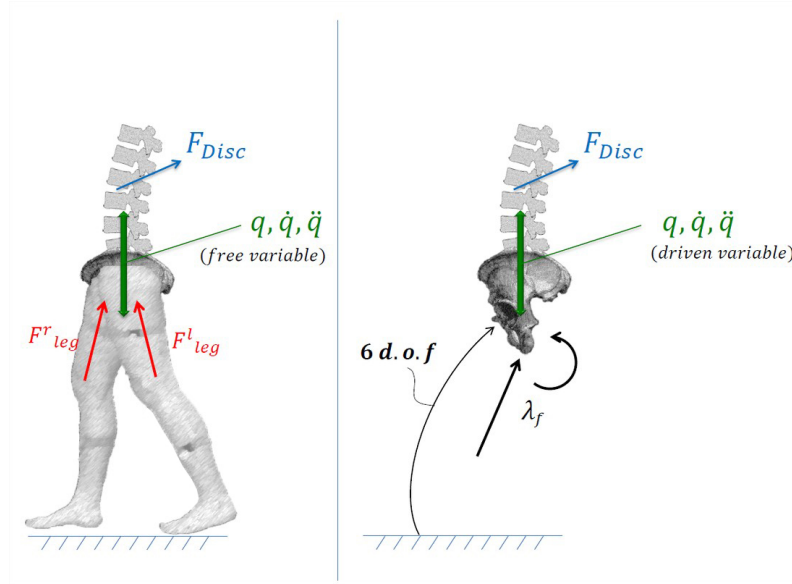


Figure 2.12: Direct dynamics approach (left). Inverse dynamics approach (right). q = position, $\dot{q}$ = speed, $\ddot{q}$ = acceleration, λ_f = constrained forces and torques [8]

The muscles modeled in this present work are two of the erector spinae muscles located in the back of the body and two of the rectus abdominis present at the front of the body. In reality, those muscles are subdivided in a series of other muscles and additional muscles that also support the spine. But for a sense of simplicity of the model and limited computation time, only 4 muscles were modeled. They only function in traction like real muscles.

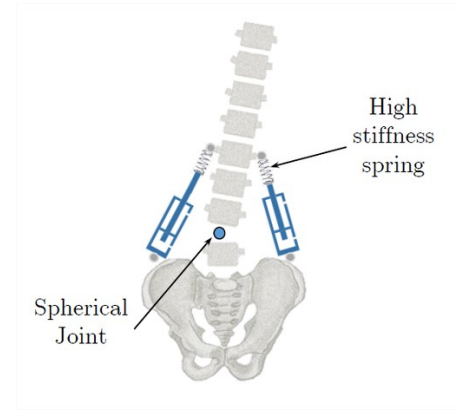


Figure 2.13: Model of the muscles [8]

A muscle was modeled as a virtual high stiffness/damping element in series. The loop was therefore closed by a force and not by a constraint, so a cut a body or joint in two was not needed. A high stiffness allows the output curve not to deviate from the imposed kinematics of the muscles. But a high stiffness would also put in difficulty the integrator since it would become a stiff problem. A stiffness value of 10^7 [N/m] and damping value of $K/10$ was sufficient for the simulations since it was shown [8] that increasing the stiffness more would not have a huge impact on the intervertebral forces. The anchoring point coordinates of the muscles on the pelvis and thorax were given by G. Abedrabbo.

The third step, since all the movements were driven, was based on the inverse dynamics approach. Thanks to the simulation conducted for step 3, the kinematics (length) of the muscles were recorded. For the 4th step, the neutral length of those muscles were set as the recorded data in the previous simulation. It's a mix of direct and inverse dynamics because at the beginning of this simulation, the movement of the spine is imposed by the muscles (direct), but after that, the efforts in the spine are computed (inverse). The changing length of the muscles is a combination between a driven variable (length of muscles changed thanks to the actuator(cylinder) in FIGURE 2.13), and a high stiffness spring inducing degrees of freedom and connecting the muscles to the spine.

An analogy can be made between the modeling of the muscles in this master thesis and two other

things: the lumped parameter viscoelastic model called the Maxwell model and series elastic actuators (SEA).

In biomechanics, there exist several models representing the viscoelastic behavior of biological tissues. One of the simplest models is called the Maxwell model (FIGURE 2.14) and is described by a spring in series with a damper.

In biorobotics, a similar analogy can be made with SEA (FIGURE 2.15). They are quite often used in this field to represent artificial muscles. It allows the motor to ensure force control by controlling the extent to which the spring in series is compressed. Moreover, it provides a protection of the motor from shocks (through the spring). Those series elastic actuators can be used from the field of a bionic ankle foot prosthesis to the one of humanoid robots.

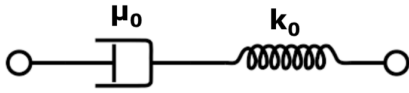


Figure 2.14: Maxwell model [54]

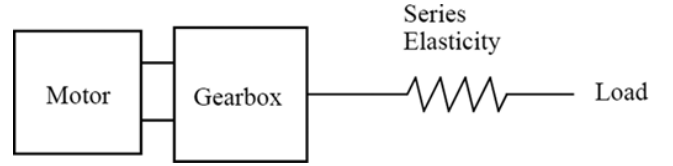


Figure 2.15: Series elastic actuator [55]

State of the joints

The only joint changing state compared to step 3 is the rotations of the thorax IMU since the thorax is not driven by the IMU anymore but by the muscles.



- ① driven (=translations via dpt points in .mbs)
- ② dependent → what we want to find
- ③ driven (=fixed to the values calculated in SECTION 2.3.1)
- ④ driven (=fixed to the values calculated in SECTION 2.3.2)
- ⑤ driven ($R_{1,2,3} = f(t)$) for the pelvis and independant for the thorax
- ⑥ dependent (on the position of the vertebrae in the space)

2.4 Computer implementation

One of the biggest part of this thesis was to start from zero and implement in C the code dealing with the model of the IMUs, the spine and the muscles in Robotran. Everything was written in C, except when stated otherwise. The input data of this computer implementation is two files coming out of the IMU: `EulerAngles.csv` (packets (time), roll, pitch, and yaw data) and `CalInertialAndMag.csv` (packets (time), magnetometer, accelerometer, and gyroscope data).

FIGURE 2.16 is the diagram representing the computer implementation of this master thesis.

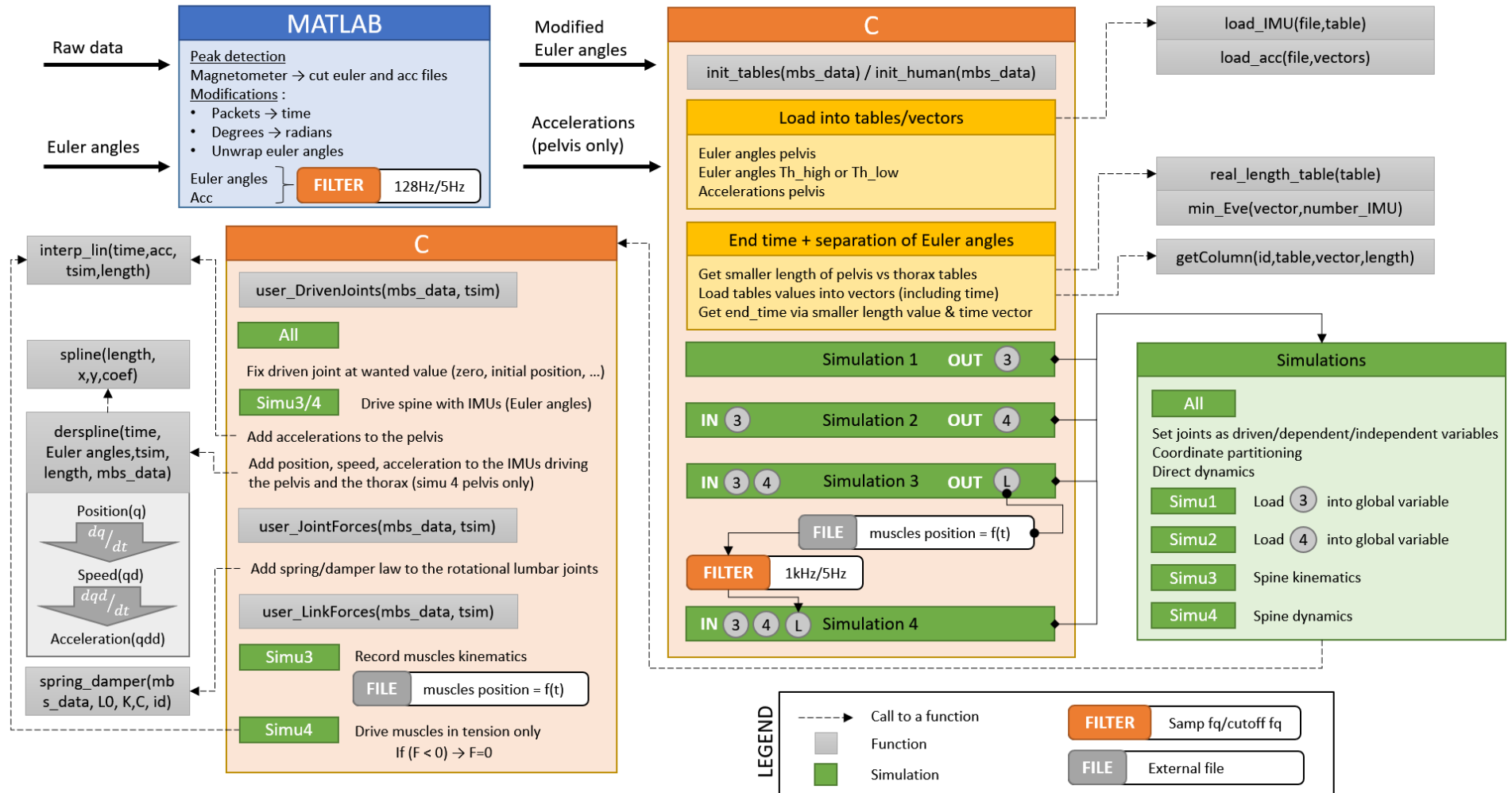


Figure 2.16: Computer implementation

2.4.1 Data processing (Matlab)

With the Bluetooth mode, synchronization of the different IMUs was easy because all the files of the IMUs started at the same time (see APPENDIX A to understand why the Bluetooth mode was not used). With the μ SD cards, a file is created when the side button is pushed. Since the side buttons of the IMUs are not pushed at exactly the same time, a way to synchronize the files had to be found. This is why a magnetic pulse was released just before each movement (for the experiment in SECTION 3.5) in a way that the IMUs were able to record that peak in their magnetometer data.

A Matlab code (APPENDIX B.1) has been written in order to detect that peak (FIGURE 2.17) of magnetism and cut the file containing the Euler angles and the accelerations accordingly.

This same code was also used to write a file that could be easily read and used by the C code surrounding Robotran to perform the analysis and simulations. This Matlab code changed the angles received in degrees and limited by $[-180^\circ 180^\circ]$ (wrapped angles) into angles in radians (since Robotran express angles in radians) and not limited by values (unwrapped angles) (see FIGURE 2.18). Indeed, Robotran needs continuous angles to work correctly. Additionally, the time available in the .csv files was in packets. Thanks to a simple formula, those packets could be transformed in seconds. Indeed, for example, when 128 packets are sent, 1 second has passed if the frequency is equal to 128 Hz.

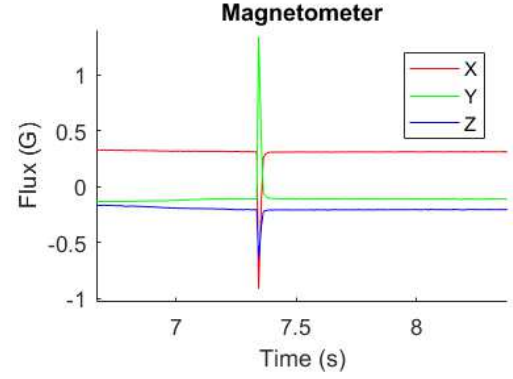


Figure 2.17: Magnetic peak in the magnetometer data of the IMU

$$time(s) = \frac{time(packets)}{frequency}$$

One could ask himself if the peak of magnetism could have an effect on the Euler angles. It did not have an effect when the experiment was done in IMU mode, thus not using the magnetometer to calculate the Euler angles. It was therefore normal that the magnetic peak did not have any influence. In the AHRS mode, a difference of 1° could be observed.

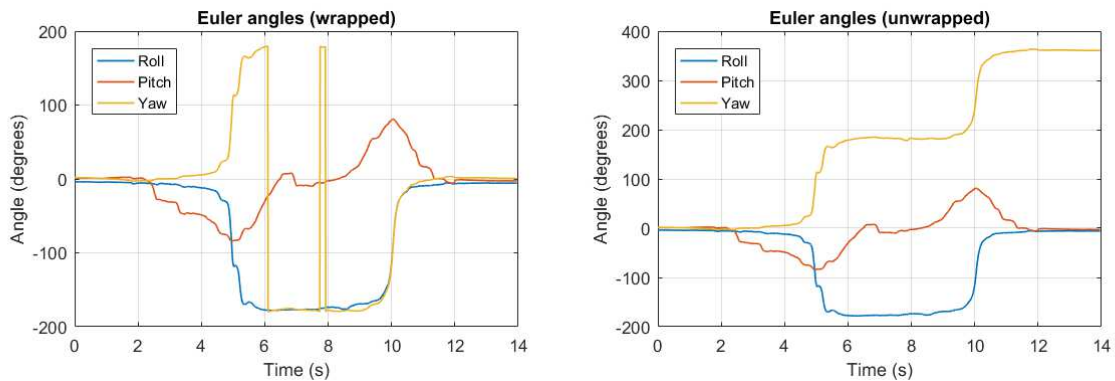


Figure 2.18: Wrapped (left) and unwrapped (right) Euler angles

As seen in FIGURE 2.16, the Euler angles and acceleration data are passed through a low-pass filter with a sampling frequency of 128 Hz (data acquisition frequency) and a cutoff frequency of 5 Hz. Indeed, it is important to filter possible noise in the input data of the simulation or the output would not be correct. [56] says that under 10 Hz, 98% of the signal remains. [57] explains that, traditionally, kinematic data are filtered at low cutoff frequencies (≈ 6 Hz). It is said in [58] that the noise is linked to the digitization

of the data. Thankfully, this noise tends to be high frequency whereas the signal like stated before is closer to low frequencies (FIGURE 2.19).

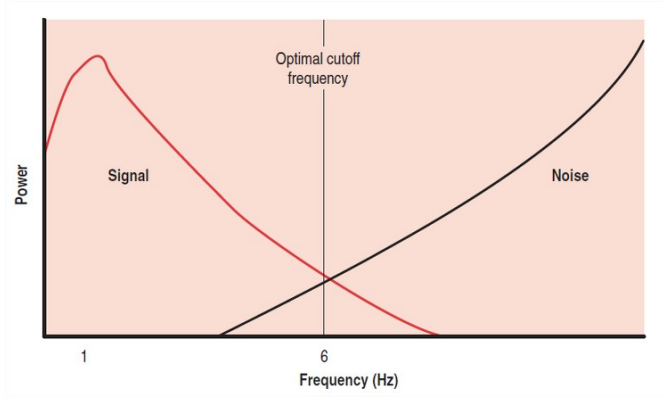


Figure 2.19: Frequency spectrum of the kinematics data [58]

Choosing the right cutoff frequency is important because if it is too low, the data is over-smoothed and if it is too high, the noise remains. The goal is therefore to have the biggest signal to noise ratio.

To find the best cutoff frequency, the frequency spectrum of the different data used in this master thesis have been studied. For a flexion of the thorax movement, the majority of the frequency spectrum of the 3 accelerations (x, y and z) was under 2Hz and under 3Hz for the Euler angles. The frequency spectrum of other movements can be seen in TABLE 2.10.

Movement	Accelerations fq (Hz)	Euler angles fq (Hz)
Flexion forward (thorax)	2	3
Flexion left (thorax)	1	2
Jump	5	5
Walk	5	3

Table 2.10: Potential cutoff frequencies for different movements

Therefore, the cutoff frequency of 5 Hz was chosen. FIGURE 2.20 shows the raw and filtered data for the forward flexion of the thorax.

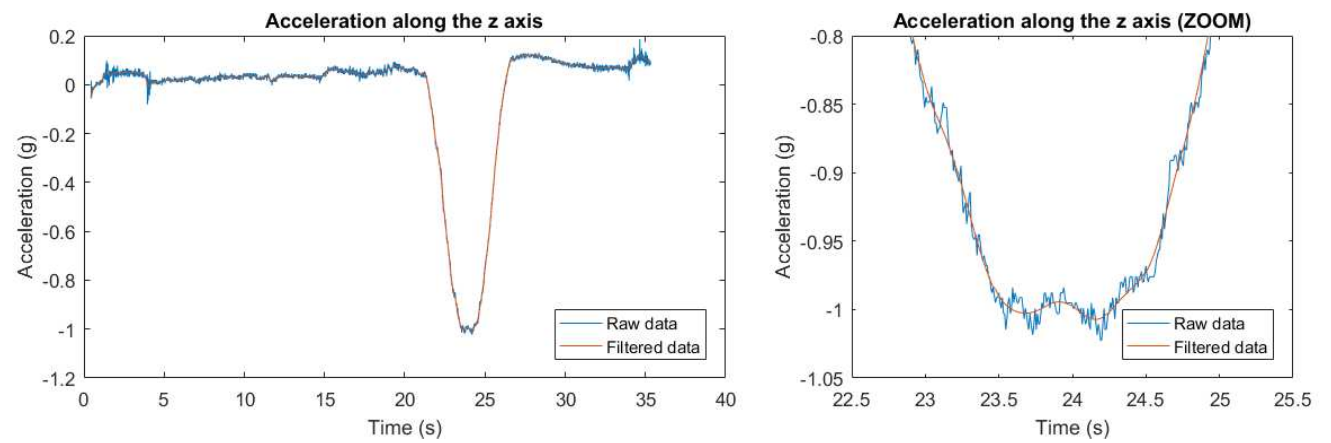


Figure 2.20: Z-acceleration (and zoom) of the pelvis IMU for a forward flexion movement

The filter chosen is a fourth order zero-phase shift low-pass Butterworth filter. Let us analyze why this filter was chosen:

- **Low-pass filter:** like stated before and represented in FIGURE 2.19, the majority ($\approx 90-95\%$) of the signal frequency spectrum is under the cutoff frequency and the noise is located higher on that spectrum.
- **Butterworth filter:** this is the most used filter in biomechanical data smoothing because it is optimally flat in its pass-band. It means that the data of interest in the pass-band is not altered by the filter. This filter also has high a roll-off and a rapid response in the time domain [cite journal]. The roll-off is the rate of attenuation of the noise above the cutoff frequency. D. Robertson [cite journal] adds that for double differentiation, Butterworth filters are the best since the amplitude of high frequency noise increases after differentiation.
- **Zero-phase shift:** time delay can happen in Butterworth filters. In order to avoid a lagging of the filtered data, it can be put through the filter twice: once in the forward direction and once in the opposite one. If the net phase shift is equal to zero, the filter is called a zero-phase shift filter.
- **Fourth order:** the highest the order, the sharper the roll-off.

2.4.2 Initialization of variables and loading of the data

The code in C starts with this section which explains the initialization of the different variables that will be needed in the program. The initialization of the anthropometric data of the human will also take place in this section.

FIGURE 2.21 shows the structure of the global variables of the program meaning the variables accessible everywhere in it.

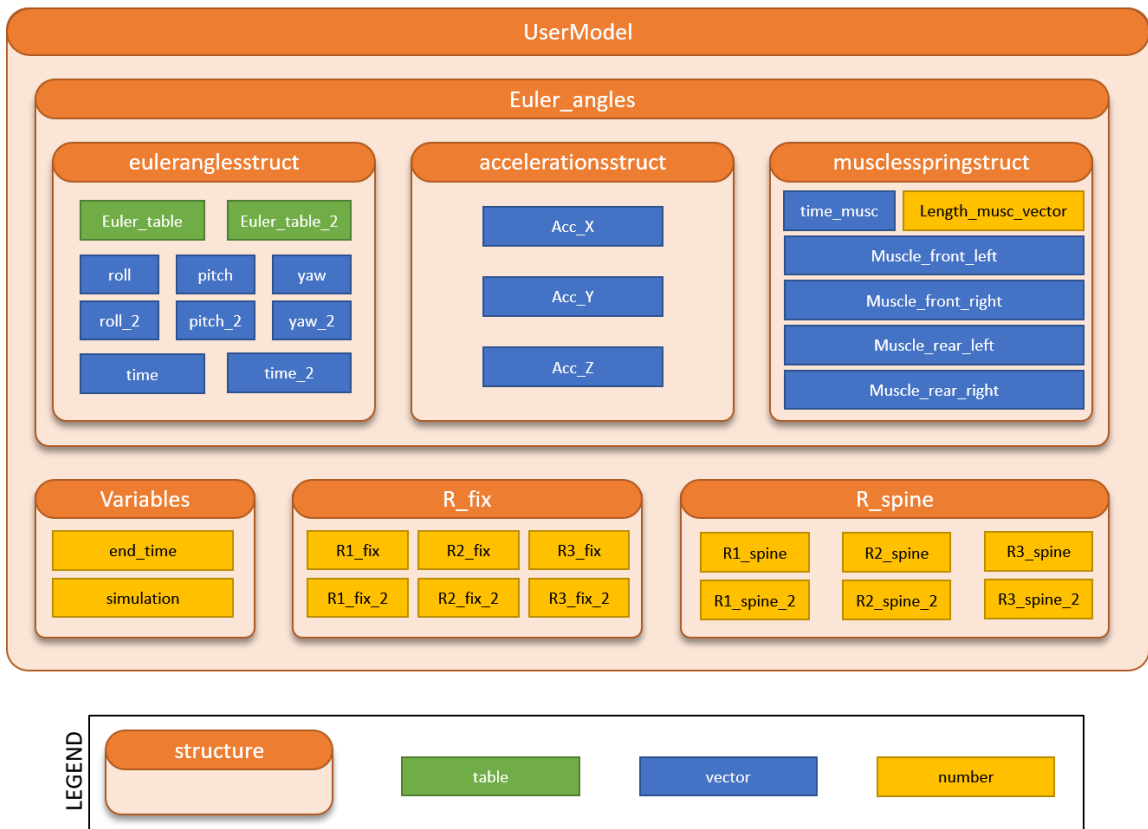


Figure 2.21: Structure of the global variables

The first step is to initialize all those values to zero, then, to take the files written by the Matlab code in SECTION 2.4.1 and store them in the two tables in the `euleranglesstruct` (pelvis and `th_low` or `th_high`) and in the `accelerationsstruct` (pelvis). `musclespringstruct` will be used later.

The tables are then separated in vectors representing the different Euler angles and time. An important step has to be executed, finding the ending time of the simulations. Indeed, two ending times are possible, the one from the pelvis file and the one from the thorax file. A simple function has been written in order to find the shortest ending time between the two vectors and store it in the global variable in the structure `Variables` (see FIGURE 2.21).

The function `init_human` is called in order to enter the masses and inertia in the Robotran model. The inertia for the lumbar vertebrae and the pelvis are calculated in this function. For the inertia of the upper part of the body, it has been computed via an external Matlab program (not shown in FIGURE 2.16). The code can be found in APPENDIX B.2.

2.4.3 Simulations

This part of the program is focused on setting the parameters for the simulations. First, for every simulation, the joints are set as driven, dependent or independent (see SECTION 2.3). After that, the coordinate partitioning step is achieved thanks to a built-in module in Robotran. This step is important when a closed-loop is present in the system. It checks that the “best” choice has been made for the independent (including driven) and dependent variables for numerical conditioning purposes mostly related to the inversion of the Jacobian matrix of the dependent variables. The reader is invited to read [31] if he wants to know more about the numerical process behind the Robotran program. After that, the direct dynamics can be realized, again, thanks to a built-in module. One could ask himself why only making direct dynamic simulations since it was said previously that inverse dynamics were also present. In reality, the direct dynamics module also implements the inverse dynamics and it is therefore easy to obtain the efforts in the joints thanks to it.

Four simulations can be seen in FIGURE 2.16 :

- **Simulation 1** refers to the first step of SECTION 2.3. The input of this simulation is the geometry of the spine. The output is the angles between the transition bodies and the pelvis/thorax. Those angles are stored in the values in the `R_spine` structure.
- **Simulation 2** refers to the second step of SECTION 2.3. The input of this simulation is the geometry of the spine, the angles computed in the first simulation, and the IMUs Euler angles data. The output is the angles between the transition bodies and the IMUs. Those angles are stored in the values in the `R_fix` structure.
- **Simulation 3** refers to the third step of SECTION 2.3 and the beginning of the fourth step. The input of this simulation is the geometry of the spine, the angles computed in the first and second simulations, and the IMUs Euler angles and pelvis acceleration data. The output is the kinematics of the spine and the muscles. Indeed, the length of the spring-damper system representing the muscles are stored in 4 external files.
- **Simulation 4** refers to the second part of the fourth step of SECTION 2.3. The input of this simulation is the geometry of the spine, the angles computed in the first and second simulations, and the pelvis IMU Euler angles and acceleration data. Moreover, the 4 files are read, stored into the `musclespringstruct`, and passed through a Butterworth low-pass filter with a sampling frequency of 1kHz ($dt = 10^{-3}$ s) and a cutoff frequency of 5 Hz (APPENDIX B.3). Those vectors are used to drive the upper part of the body. The output is the dynamics of the spine.

Each time a new simulation starts, the value *simulation* in the structure `Variables` is changed. It will be used to recognize in which simulation the program is in the user_functions (see SECTION 2.4.4).

2.4.4 User functions

During the simulation process, user functions are called at each time step. Joint forces, external forces, link forces and the motion law of the driven joints are implemented in these functions. This master thesis does not deal with external forces so this function will not be used.

`user_DrivenJoints(mbs_data, tsim)`

In this function, for all the simulations, the driven joints are fixed to their wanted value (zero, initial values and positions, etc.). For the third and fourth simulations, accelerations are added to the pelvis. A linear interpolation allows us to get a value of the acceleration for each `tsim` (= current simulation time). For those two simulations, the spine is driven by the IMUs (pelvis only for simulation 4). Therefore, the Euler angles of the IMUs need to be given a value for each `tsim`. Moreover, the speed and acceleration of those Euler angles need to be computed. The function `derspline` deals with that. It stands for derivative and spline. Indeed, the most challenging part is to translate the Euler angles as a function of the time in a spline, derivate it two times and get the value of the speed, and acceleration of those angles as a function of the simulation time (`tsim`).

First, a third order interpolation (FIGURE 2.22) needs to be implemented in order to obtain a usable acceleration. Indeed, after two derivations, the acceleration cannot have discontinuities.

$$\begin{cases} \text{Position} & q = ax^3 + bx^2 + cx + d \\ \text{Speed} & \dot{q} = 3ax^2 + 2bx + c \\ \text{Acceleration} & \ddot{q} = 6ax + 2b \end{cases}$$

This interpolation needs to have as output the coefficients a , b , c , d of the position in order to derivate manually³ the position to obtain the speed and acceleration of the Euler angles. The function `spline` takes as input the vectors x (time), y (Euler angle) and their length. It follows the algorithm stated in [59]. The code was first copied from [60] but it did not work so it had to be changed a little to match the program.

After obtaining the coefficients for the position, speed and acceleration of the Euler angles, they are simply, thanks to a scalar product, multiplied to various powers of `tsim` ($=x$ in the q , $\dot{q}$, and $\ddot{q}$ formulae). The codes for the `spline` and `derspline` (derivates of the spline) functions can be found respectively in APPENDICES B.5 and B.4.

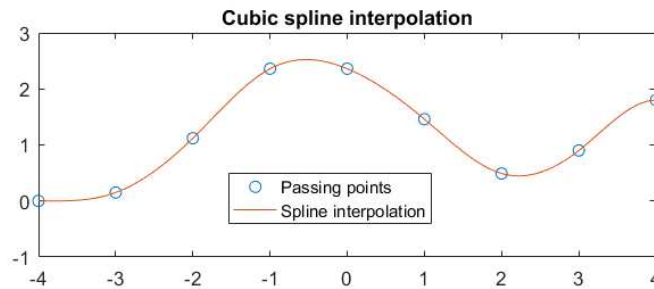


Figure 2.22: Cubic spline interpolation visualization

³Manually means that since the spline is a polynomial, it can easily be derivated by multiplying the coefficients by constants like stated in the q , $\dot{q}$, and $\ddot{q}$ formulae.

user_JointForces(mbs_data, tsim)

This functions adds a spring-damper law to the rotational lumbar joints.

$$F = -(K (\theta - \theta_0) + C \cdot \dot{\theta})$$

with K , the stiffness of the joints (see TABLE 2.1) and C the damping coefficient ($K/8s$).

user_LinkForces(mbs_data, tsim)

This function deals with the forces in the muscles. In simulation 3, it is used to record the length of the muscles and to store them in 4 external files. In simulation 4, it is used to transform those values (put in vectors at the start of the simulation) into a single value for each tsim thanks again to a linear interpolation function.

In simulation 4, once the value of the length of the muscle for every tsim is known, the law of a spring-damper system can be applied.

$$F_{link} = K_{muscle} \cdot (Z - Z_0) + C_{muscle} \cdot Z_d$$

with Z_0 , the neutral length of the spring-damper system, Z its true length, and Z_d its speed.

The muscles only work in tension and it is known that a negative force means that the spring is in compression and a positive force means that it is in traction. Therefore, when the force is smaller than zero, it is equaled to zero.

2.4.5 Graphs (Matlab)

All the analysis are done thanks to graphs on Matlab. Indeed, it is an easy tool to draw graphs and compare values. Making graphs in C requires a graphic interface that is hard to code. The values can be retrieved from files created by Robotran or created by the user thanks to a built-in function in C called `system_output`.

This chapter focuses on the experiments realized within the framework of this master thesis. SECTION 3.1 describes the choice of the data acquisition mode and algorithm of the x-IMU. The experiment in SECTION 3.2 was carried out in order, first, to observe the effects of a change of inertia of body segments on the intervertebral efforts of the lumbar spine. The second goal of this experiment was to be able to choose the best set of anthropometric data for the model presented in CHAPTER 2. SECTION 3.3 explains why the results of the third simulation should not be used in the analysis of intervertebral efforts and the differences between the third and fourth simulation results. SECTION 3.4 analyzes the noise present in the intervertebral efforts in order to find its source. SECTION 3.5 finishes this chapter by verifying the rigid thorax hypothesis and by analyzing the intervertebral efforts for different daily life movements.

3.1 Choice of the x-IMU mode/algorithm

3.1.1 Introduction

The first question which arises is how the IMUs will be used. This section focuses on the choice of the IMU mode and the algorithm used. Indeed, the previous master thesis [33] did not state the mode or algorithm used for their experiments.

First experiment : x-IMU mode

As stated in FIGURE 1.5, there are two modes to the IMU. It has to be chosen before the acquisition of the data. SECTION 1.3.1 explains the difference between the IMU mode and the AHRS mode.

The goal of this first experiment is to find out if the drift (error) is lower in IMU mode or in AHRS mode. Indeed, the error should be the lowest possible since the movements of the spine are only a few degrees. The hypothesis is that the AHRS mode should be more precise since there is a recalibration of the data via the magnetometer values that are not used in the IMU mode.

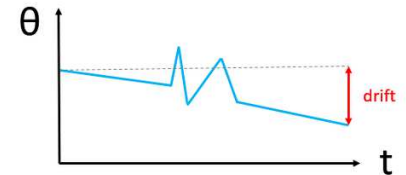


Figure 3.1: Drift

Second experiment : algorithm

SECTION 1.3.1 explains that the data acquisition can be done either with the embedded algorithms (IMU or AHRS) or external algorithms (Mahony or Magdwick). This experiment checks if the delay is lower when the embedded algorithm in the x-IMU is used, getting the Euler angles immediately, or when the raw data of the sensors is taken and the Mahony algorithm or the Magdwick algorithm is applied on it, retrieving the Euler angles wanted. It's important to notice that the x-IMU can be in either IMU or AHRS mode if the external algorithms are used. It does not matter because the raw data is taken for the external algorithms.

The delay should be the lowest possible for the experiments and as responsive as possible, in order to avoid errors in the measures. The hypothesis is that the delay should be lower with the Madgwick algorithm because it is an evolution of the Mahony algorithm.

The external algorithms were given by the official site of the *x-IMU* [46].

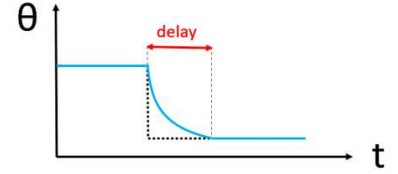


Figure 3.2: Delay

3.1.2 Materials and methods

Materials

- 1 x-IMU [46]
- 3 fixed perpendicular surfaces (forming a corner)
- 1 computer with a Bluetooth connection
- x-IMU GUI software [46]

Methods

First part : collecting the data To begin this experiment, the x-IMU board was switched on thanks to the on/off button on its side. After that, it was connected to the Bluetooth of the computer thanks to the Windows interface. The user interface (x-IMU GUI) was used to open the port of the computer to be able to read the data sent by the board. For the first experiment, the register containing the algorithm mode was written to AHRS (magnetometer used) via the user interface. For part of the second one, it was written as IMU (only gyroscope and accelerometer used). The acquisition frequency was chosen to be 128 Hz.

Next, the board was placed against the 3 fixed surfaces like shown in FIGURE 3.4 and oriented like stated in FIGURE 3.3.

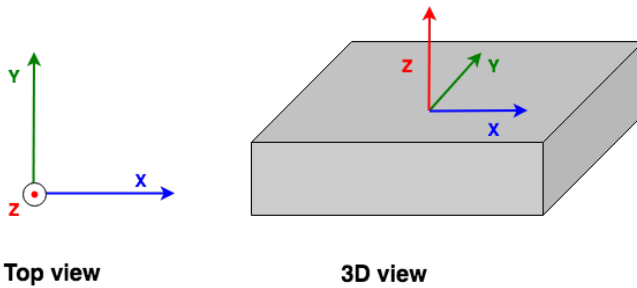


Figure 3.3: Start of the experiment: orientation of the x-IMU



Figure 3.4: Start of the experiment: setting of the x-IMU

The logging of data was initiated in the user interface on the computer then the board was not moved for approximately 30 seconds. Indeed, the external algorithms needed to converge to the angle value (FIGURE 3.6) and they did not have two different gains, in contrast with the internal algorithms which had two gains in order to reach the right value fast at the beginning with a high gain and avoid the overshoots with a lower gain during the movement.

First experiment After the wait, the board was rotated by hand around its Z-axis.

Second experiment After the wait, the board was rotated by hand around one of its axis (e.g. X-axis) for ≈ 4 turns.

It was then repositioned exactly against the 3 surfaces in the same orientation as the starting point of the experiment. Finally, a wait of approximately 30 seconds was implemented to leave time for the curve to steady itself and tend to its final value. The logger was stopped then thanks to the user interface and the file was stored.

The second experiment was repeated twice per axis and an additional random movement was also realized twice. The total number of files stored is therefore of 9 (8+1).

Second part : reading the data The files obtained thanks to the user interface were in the file format `.csv`. Since Matlab [61] was the software used in this master thesis at the beginning, a conversion needed to be applied from `.csv` to `.mat`, a file containing variables and tables easily read by Matlab.

As soon as the `.mat` files were created, they could be inserted into the larger code (given by [46] and adapted by the user) used to read them and create graphs. Values for the gains were chosen in order to reach the wanted value before the movement started but in a way that there was as little overshoot as possible in the movement.

This code does 3 things. It reads and plots the Euler angles given :

- directly by the IMU
- by the Madgwick algorithm
- by the Mahony algorithm

First, the code reads the `.mat` file. Then, it plots the raw data, meaning the gyroscope, accelerometer and magnetometer entries ((x, y, z) vectors). After that, it processes the sensor data through the Madgwick algorithm and in parallel does the same with the Mahony algorithm, getting the Euler angles given by the two algorithms and plotting them. Finally, it plots the comparison between IMU, Madgwick and Mahony for each angle (roll, pitch, yaw).

3.1.3 Results and discussion

Results

First experiment (mode) FIGURE 3.5 shows the results obtained during the first experiment. This graph was produced by turning the board around its Z-axis.

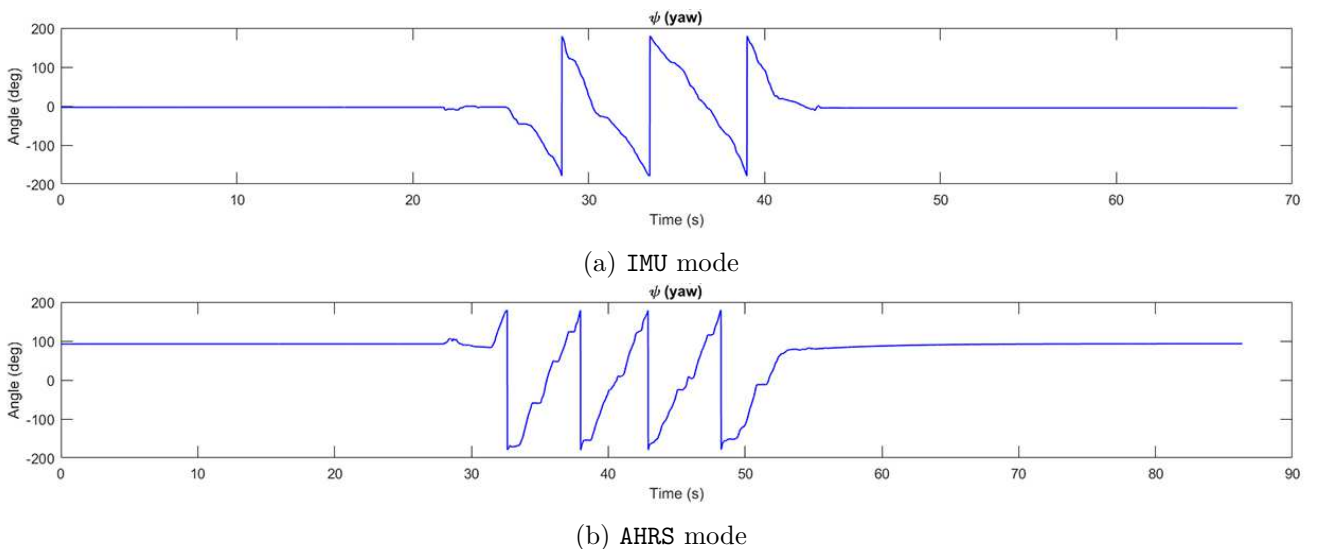


Figure 3.5: Euler angles comparison (Z-axis)

TABLE 3.1 presents the drift obtained for the IMU and AHRS modes.

Error [deg]	Without Magnetometer	With Magnetometer
Roll	0.025	0.026
Pitch	0.015	0.062
Yaw	1.830	0.440
Total	1.870	0.528

Table 3.1: Errors for the drift : IMU mode (left) and AHRS mode (right)

Second experiment (algorithm) FIGURE 3.6 shows one of the results obtained during the second experiment. This graph was produced by turning the board around its X-axis. This experiment was conducted with the AHRS mode. So when it is stated IMU, it means the IMU internal algorithm.

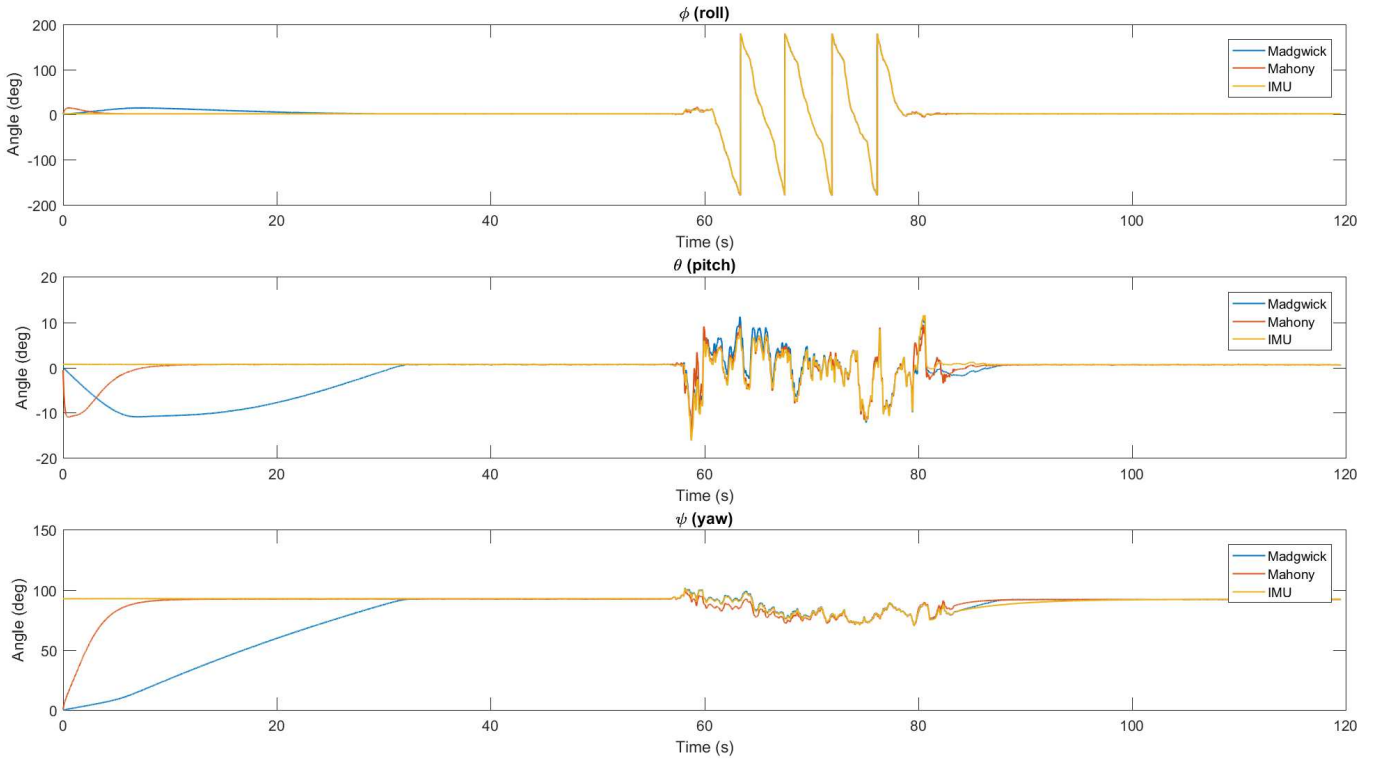


Figure 3.6: Euler angles comparison (X-axis)

TABLE 3.2 presents the delay obtained for the different algorithms : the embedded algorithm in the IMU, the Magdwick algorithm and the Mahony algorithm. The X field represents the mean of the two experiments where the IMU was rotated around its X-axis. The same goes for Y, Z and Random.

Delay [sec]		IMU	Magdwick	Mahony
X	Roll	7.120	2.930	2.930
	Pitch	5.220	6.320	7.390
	Yaw	20.280	6.210	8.685
Y	Roll	2.150	3.690	2.290
	Pitch	4.530	5.920	5.985
	Yaw	18.345	6.405	7.420
Z	Roll	2.560	5.360	5.835
	Pitch	4.345	5.955	7.085
	Yaw	19.370	6.635	8.230
Random	Roll	5.295	5.160	4.835
	Pitch	3.235	2.990	4.325
	Yaw	19.645	5.535	8.900
Total		9.341	5.259	6.159

Table 3.2: Delay for the different algorithms

Discussion

First experiment : x-IMU mode

It can be seen as expected that the yaw drift is bigger in the experiment without the magnetometer. FIGURE 3.5 shows only the yaw angle because it is the more important since it is the one which is not recalibrated, in the IMU mode, with the magnetometer value.

The results of this experiment need to be taken with a grain of salt. Indeed, it was based on only two tests. Moreover, it can be seen on the AHRS graph that the x-IMU was rotated 4 times and on the IMU graph that it was rotated only 3 times. Let us emphasize the fact that this experiment and the next one were realized against a wall and by turning the x-IMU by hand. It was therefore not a professional experimental setup.

The previous master thesis [33] experimented the effect of the magnetic field of the treadmill on the Euler angles and found that it was best to limit the external magnetic fields addition to obtain the more precise measures as possible. They also found that the fastest the IMU was rotating, the more imprecise were the measures.

Second experiment : algorithm

The values in TABLE 3.2 seem to be enormous but they are computed for a difference of 0.05 degrees to the final value. In reality, it can be seen in FIGURE 3.6 that the movements are followed quite well.

The yaw angle delay is the biggest in the table for the internal algorithm. A question that can be asked is, is the yaw angle delay really important? The two other angles are more important concerning the spine. Indeed, the movements a person does during the day are rarely movements of torsion but

more of flexion/extension and lateral flexion. Moreover, even if the yaw angle delay at the end of the movement is quite large, the delay during the movement seems to follow the other curves very closely.

Madgwick seems to be the best algorithm but the signal is extremely noisy (see FIGURE 3.7). Mahony is a little less noisy but the best signal is clearly the IMU signal.

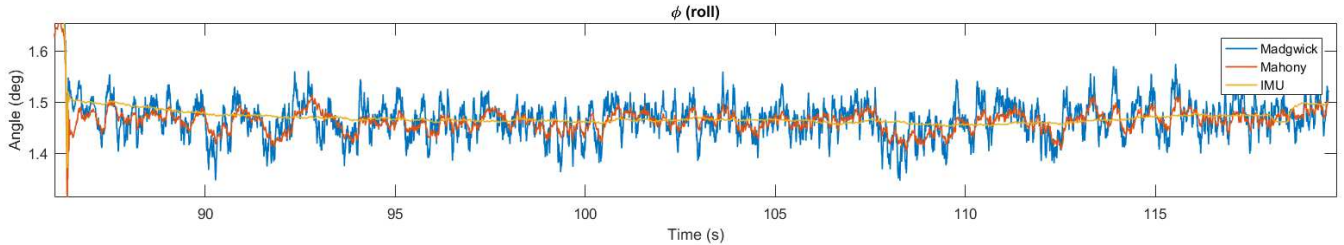


Figure 3.7: Noise

Therefore, compromising between noise and accuracy is important. Moreover, there seem to be less overshoots in the IMU curve. Additionally, the IMU data has the advantage to settle fast to the wanted value thanks to its double gain: one high gain to settle fast to that value and one smaller to avoid overshoots. All of these reasons leads us to choose the internal algorithm of the IMU.

3.1.4 Additional comments

This experiment has been conducted with Bluetooth and via Matlab, but it was done at the start of the year. Later in the year, it was chosen to pass from Matlab to the C language. In addition to the fact that IMUs were not always available when I needed them and were available only for a short amount of time, problems with the program to record several IMUs at the same time in Bluetooth occurred. Indeed, the data coming out of the IMU had steps in it, making the processing of the data impossible. Therefore, the Bluetooth mode was discarded and μ SD cards were used. Those cards were less reliable than the Bluetooth mode because the internal frame of the IMU was not the one shown on the IMU case (= frame for the Bluetooth mode). Moreover, sometimes the μ SD cards were not written (rare) and, in the AHRS mode, a few IMUs (not all) had deviations of 50 degrees (random number for each IMU) only after a few seconds. Therefore, the IMU mode was chosen. More information can be found in APPENDIX A.

3.2 Inertia effect on intervertebral disks efforts

3.2.1 Introduction

The purpose of this experiment was, first, to observe if a change of inertia of the upper part of the body, the lumbar region or the pelvis had an effect on the intervertebral disks efforts. The second goal of this experiment was to be able to choose the best set of anthropometric data for this specific model since problems initially occurred with the inertia computed in SECTION 2.1. TABLE 3.3 is copied from TABLE 2.5 as a reminder for the initial inertia computed thanks to literature.

Bodies of interest	$I_{\hat{X}} [kg\ m^2]$	$I_{\hat{Y}} [kg\ m^2]$	$I_{\hat{Z}} [kg\ m^2]$
Additional body	0.566100	0.370600	0.254000
Lumbar vertebra	0.004225	0.004225	0.008204
Pelvis	0.043130	0.037176	0.045350

Table 3.3: Inertia of the bodies of interest (with respect to their CM) for a 56 kg female human body

3.2.2 Materials and methods

Materials

For this experiment, the complete model of the spine and its IMUs (Robotran) is needed with its associated C code (Visual Studio). Some experimental data is also needed to test the model. The flexion movement was chosen for the IMU data input (Euler angles (thorax low, pelvis) and pelvis acceleration). Any other movement could have been chosen. The full methodology for the data acquisition will be explained in SECTION 3.5 but this is not the subject of this experiment.

Methods

The function `init_human` was modified in order to change the inertia values of the upper part of the body (ratio x5 and x10 compared to TABLE 3.3), the lumbar region (ratio x20, x50 and x100) and the pelvis (ratio x10 and x50). Lower ratio would lead to a mechanical problem like stated in SECTION 2.1.3.

3.2.3 Results and discussion

The results are only shown for the S1-L5 intervertebral disk since the other disks have a similar behavior.

A low-pass digital Butterworth filter with a cutoff frequency of 5Hz and a sampling frequency of 1kHz (dt simulation = 10^{-3} s) was applied on the graphs for a better readability due to the high frequency noise.

Pelvis inertia modification The ratio for the upper body is x10 and for the lumbar spine it is x50.

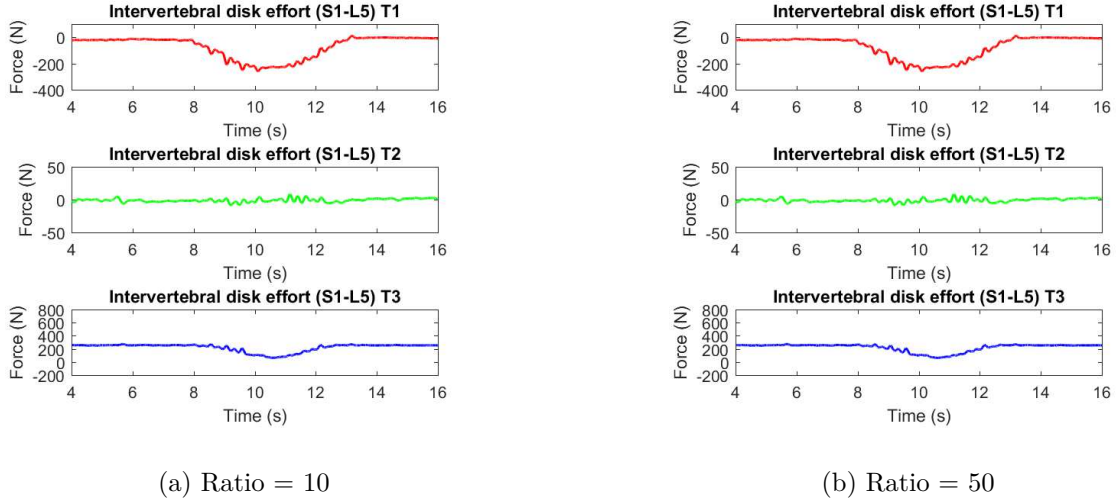


Figure 3.8: Pelvis change

FIGURE 3.8 shows that a change from a ratio x10 to a ratio x50 with respect to the TABLE 3.3 values of the pelvis inertia does not change the efforts in the intervertebral S1-L5 disk. Indeed, the maximal effort along T1 is equal to -240 N for both ratio. The effort along T2 oscillates around the zero value. For T3, the minima value around 10.7 seconds is equal to 65.3 N. Therefore, to be as close to reality as possible, the ratio x10 will be chosen for the pelvis.

Upper body inertia modification The ratio for the lumbar spine is x50 and for the pelvis it is x10.

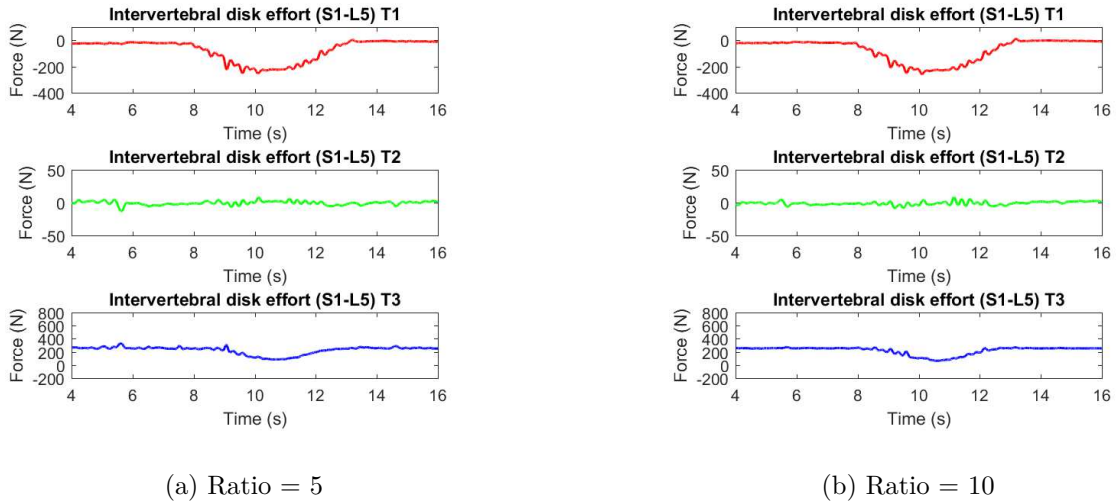


Figure 3.9: Upper body change

FIGURE 3.9 shows however that a modification of the upper body inertia from a factor x5 to a factor x10 does not change the values but helps with the sensitivity to low frequency noise (more information on the noise in SECTION 3.4), therefore smoothing the curve. It is especially noticeable in the T3 graph before and after the movement of flexion. The person is standing still so the value should remain as stable as possible. Therefore, the ratio x10 will be chosen for the model.

Lumbar spine inertia modification The ratio for the upper body is x10 and for the pelvis it is x10.

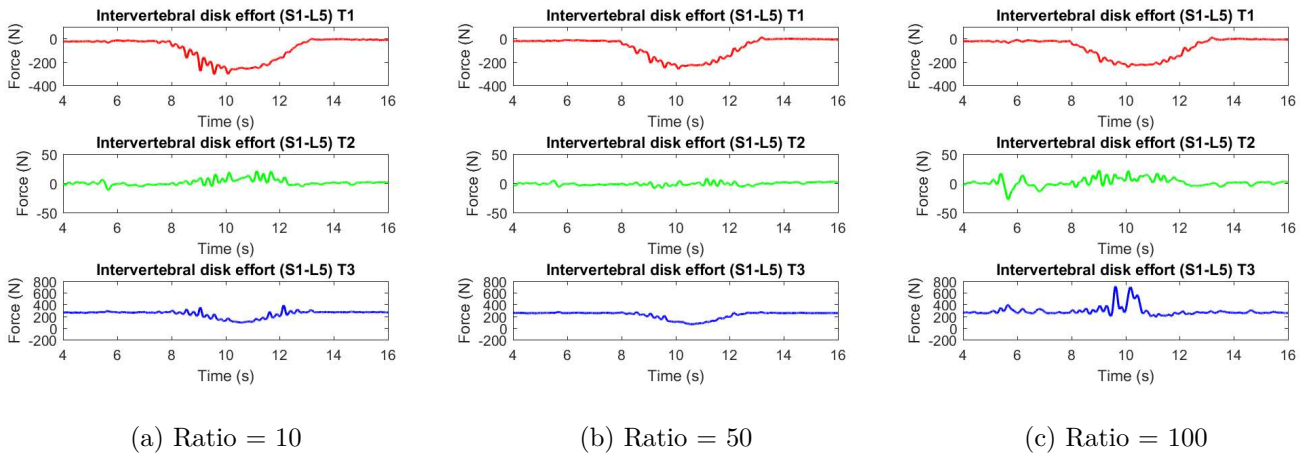


Figure 3.10: Lumbar spine change

The initial thought would be that the inertia parameter that most affects the intervertebral efforts is the inertia of the lumbar region since the principal movement occurs in those vertebrae. FIGURE 3.10 comforts that idea. FIGURE 3.10a shows that a ratio of x10 tends to make the model more subject to the low frequency noise than a ratio of x50 (FIGURE 3.10b). Indeed, a bigger inertia means that the body tends to resist to changes. FIGURE 3.10c shows that a too high inertia induces a lower frequency for the body. This low frequency adds itself to the low frequency noise and the values obtained are therefore skewed. Therefore, the ratio x50 will be chosen for the vertebrae inertia of the lumbar spine.

3.3 Thorax driven by IMU vs thorax driven by muscles

3.3.1 Introduction

SECTION 2.3.4 explained that the efforts retrieved from the third simulation - thorax driven by an IMU - were not representative of the reality and that there were differences between that simulation and the one where the spine is driven by muscles. This section dives into the analysis of the differences and their sources.

3.3.2 Materials and methods

Materials

For this experiment, the complete model of the spine and its IMUs (Robotran) is needed with its associated C code (Visual Studio). Some experimental data is also needed to test the model. The flexion movement was chosen for the IMU data input (Euler angles (thorax low, pelvis) and pelvis acceleration). Any other movement could have been chosen. The full methodology for the data acquisition will be explained in SECTION 3.5 but this is not the subject of this experiment.

Methods

The efforts were plotted in Matlab for the third and fourth simulation.

3.3.3 Results and discussion

The results are shown for the S1-L5 intervertebral disk, the other disks experiencing the same behavior.

Results

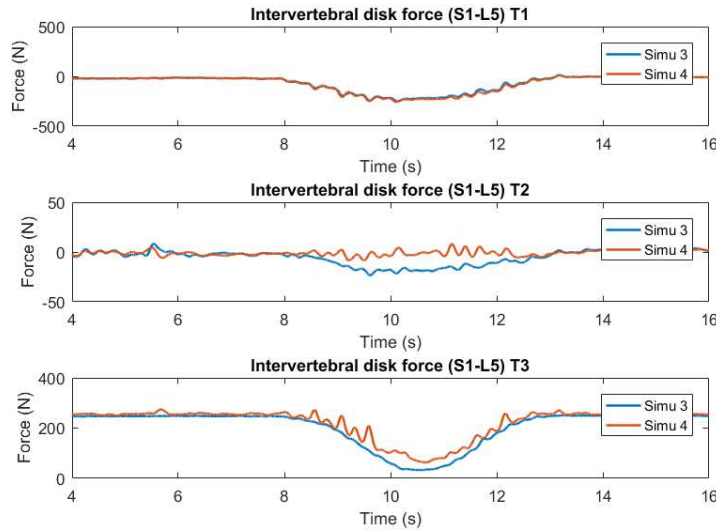


Figure 3.11: Simu 3 vs 4

Without taking into account the peaks of noise (more information about that in SECTION 3.4), the T1 curves in FIGURE 3.11 have the smallest difference with an average less than 10 N between the two curves. The highest difference occurs at 11.7 seconds with a difference of ≈ 30 N (simu 3 = -131 N and simu 4 = -161.8 N). For the T2 curves, they have a difference of 18 N (at 10.6 s) and stay close to the line of symmetry (0 N). The T3 curves have the biggest difference with a value of ≈ 64 N at 10.1 s (simu 3 = 36.33 N and simu 4 = 100.4 N).

Discussion

FIGURE 3.11 shows that the biggest change between both simulations occurs in the T3 direction which is the direction normal to the vertebra ($N_{a,b,c}$ in FIGURE 3.12). The fourth simulation, with the muscles, shows that during the movement, the effort curve does not diminish as much as in the third simulation with both IMUs.

This can be explained by a simple analogy of a block on a slope. The block represents the spine while the slope and the spring represent respectively the flexion of the thorax movement and the back muscles. When there is no slope (FIGURE 3.12a), the normal force N_a is equal to the resultant R_a , assuming that there is no tangential force. Indeed, the resultant force is equal to the the vectorial sum of the normal and the tangential forces.

$$\vec{R}_i = \vec{N}_i + \vec{T}_i \quad (\text{for } i = a, b, c)$$

When an object is laid on a slope (FIGURE 3.12b), the resultant R_b is still equal to the previous one R_a because the mass of the block did not change ($R_b = R_a = -mg$), but the normal to the ground has its value diminished. Indeed, if R_b stays the same and T_b increases due to the friction of the object with the ground, N_b has to diminish. That is why, in FIGURE 3.11, the T3 value diminishes during the movement for both the third and fourth simulation.

In FIGURE 3.12c, the back muscles add additional forces when they pull on the spine. Therefore, the normal force N_c increases as well. That is why, in FIGURE 3.11, the T3 curve of the fourth simulation is higher than in the third simulation.

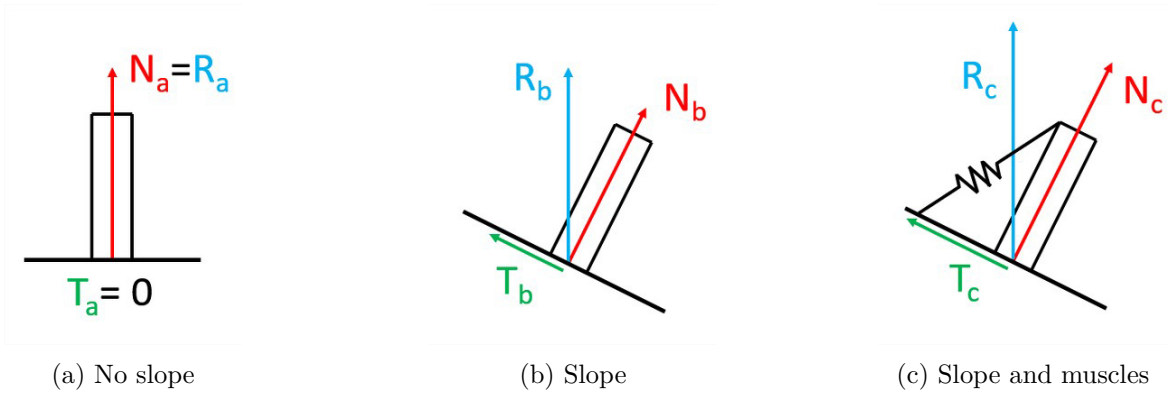


Figure 3.12: Analogy of a block on a slope

G. Abedrabbo [8] reports that “*in addition to the increased intervertebral forces, there is a significant change in the way forces are distributed along the spine. This means that a simplified model does not only underestimate those forces, but also shows a skewed behavior compared to a physiological model*”. This is why the results of the fourth simulation are the ones analyzed in this master thesis.

3.4 Analysis of the noise

3.4.1 Introduction

The intervertebral disks forces are subject to noise as can be see in FIGURE 3.13. The goal of this section is to find the origin of this noise and to propose improvements that could be put in place in order to diminish this noise.

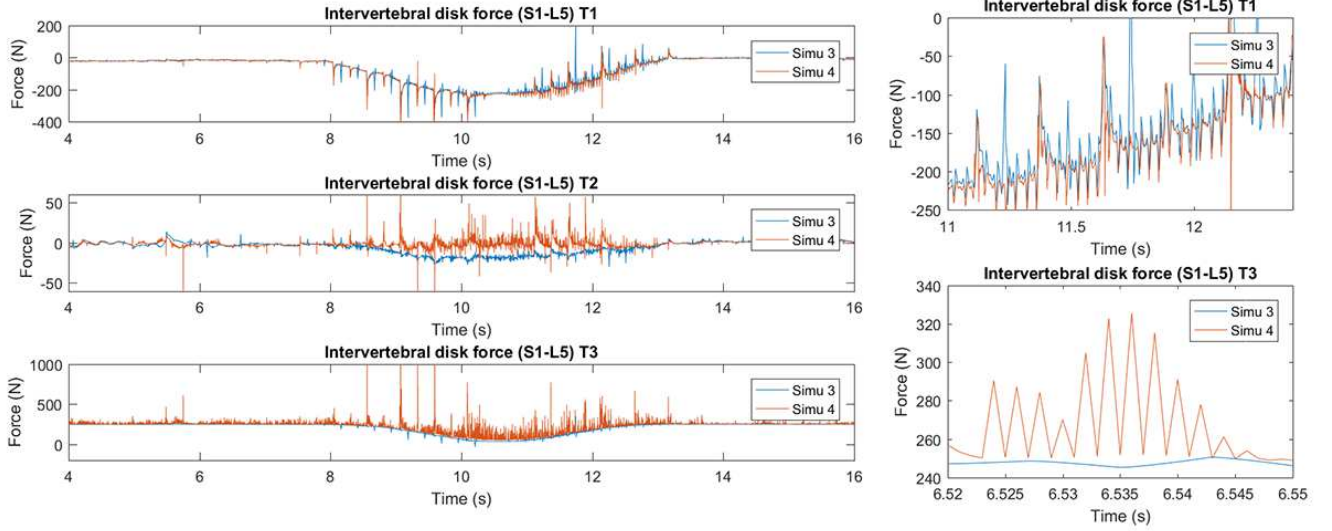


Figure 3.13: Forces in the S1-L5 intervertebral disk for a forward flexion movement

In FIGURE 3.13, for the simulation with the muscles (simu 4), 3 major noise frequencies can be observed.

Here is the formula to calculate them:

$$\text{Frequency} = \frac{\text{number of cycles}}{\text{delta time}}$$

$$f_1 = \frac{4}{12.14 - 11.12} = 3.92 [Hz] \approx 4 [Hz]$$

$$f_2 = \frac{5}{12.11 - 11.9} = 23.81 [Hz] \approx 24 [Hz]$$

$$f_3 = \frac{10}{6.544 - 6.524} = 500 [Hz]$$

The first obvious observation is that the curves for the simulation with muscles has a higher frequency noise spectrum than the curves for the spine driven only by IMUs (simu 3). The second observation is that f_1 for the simulation with the IMUs is twice the value of f_1 for the fourth simulation (FIGURE 3.14).

Some hypotheses can therefore be made :

- The muscles induce a high frequency noise (f_3) in the frequency spectrum of the signal. This noise can easily be filtered by a low-pass filter.
- The low frequency noise (f_1) could come from the Euler angles of the IMUs. Indeed, with two IMUs, the third simulation has a frequency twice the frequency of the fourth simulation who only has one IMU (pelvis). This noise could come from the spline function who interpolates the IMU data.

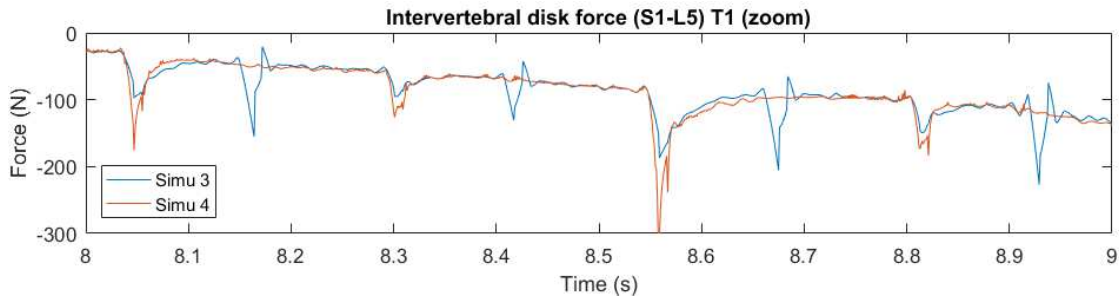


Figure 3.14: Force in T1 in the S1-L5 intervertebral disk for a forward flexion movement (zoom)

3.4.2 Materials and methods

Materials

For this experiment, the complete model of the spine and its IMUs (Robotran) is needed with its associated C code (Visual Studio). Some experimental data is also needed to analyze the noise of the model. The flexion movement was chosen for the IMU data input (Euler angles (thorax low, pelvis) and pelvis acceleration). Any other movement could have been chosen. The full methodology for the data acquisition will be explained in SECTION 3.5 but this is not the subject of this experiment.

Methods

The method is to retrace the steps from the input of the program to the output and check the obtained curves, especially in the spline function and the muscles.

3.4.3 Results and discussion

The results are only shown for the S1-L5 intervertebral disk since the other disks have a similar behavior.

Results

The encircled defects can be seen in all the IMUs data. The data of the IMU that was attached on the lower part of the thorax was chosen for the clarity of its graph.

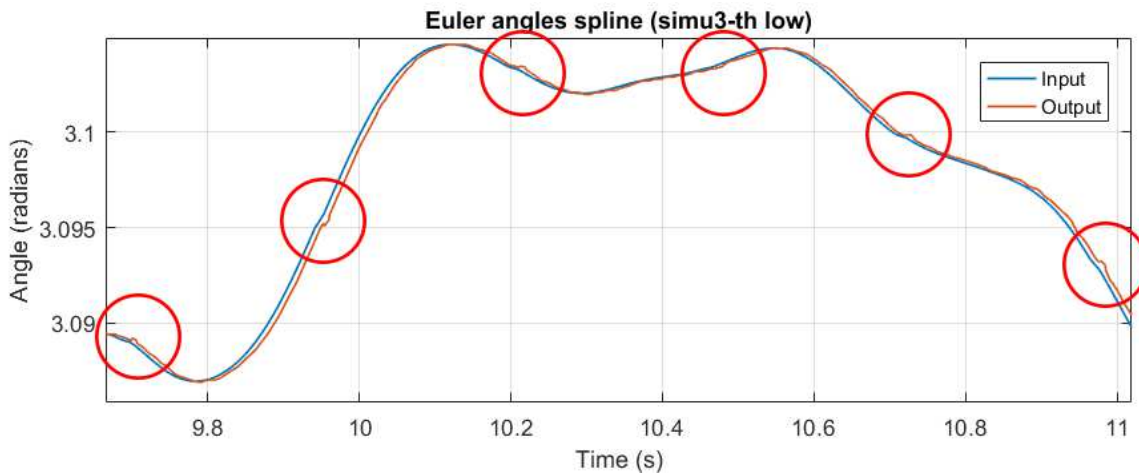
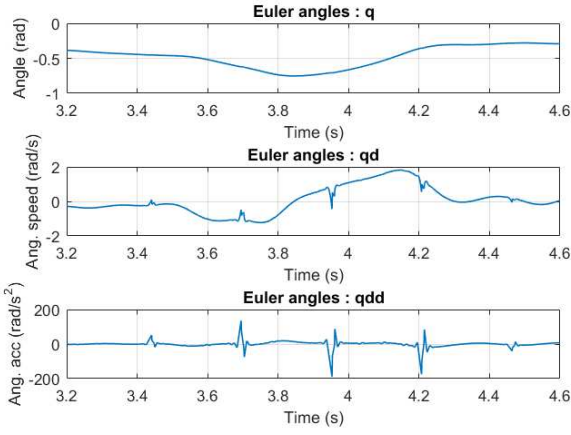
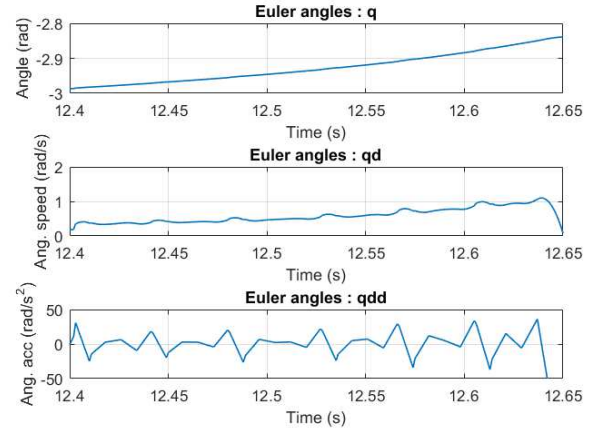


Figure 3.15: Roll of the lower thorax IMU : input and output of the spline function



(a) 3.2 s < t < 4.6 s (3.94Hz noise)



(b) 12.4 s < t < 12.65 s (25Hz noise)

Figure 3.16: Roll of the pelvis IMU (after derspline function) : angle, angular speed, angular acceleration

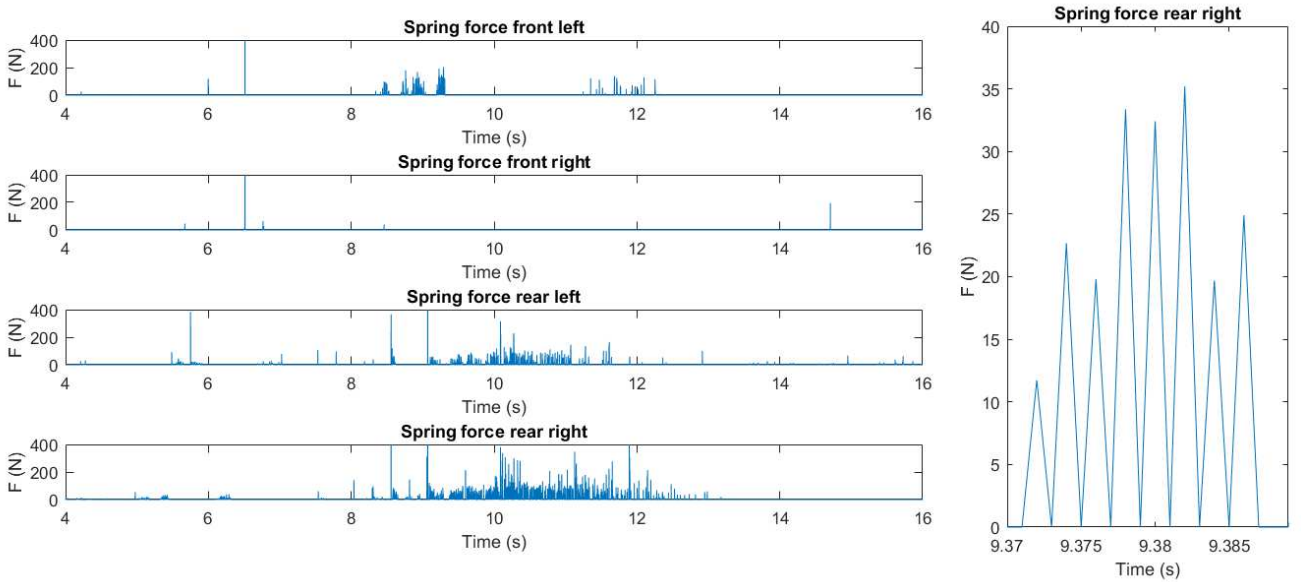


Figure 3.17: Spring forces and zoom (500Hz noise)

Discussion

4 Hz noise In FIGURES 3.15 and 3.16a it can easily be observed that a 4Hz noise is present because of the spline function. This small noise would not be a major problem if the angles were not being derivated. Because of those small defects, peaks of 200 [rad/s^2] in the acceleration are present, therefore generating peaks of forces in the intervertebral efforts (FIGURES 3.13 and 3.14).

$$f_1 = \frac{4}{4.464 - 3.45} = 3.94 [Hz] \approx 4 [Hz]$$

8 Hz noise (Figure 3.14) The difference between simulation 3 and simulation 4 can effectively be explained by the fact that simulation 3 uses two IMUs to drive the spine and simulation 4 only uses one (pelvis). It was observed in figures (not shown here) similar to FIGURE 3.15 that the pelvis had defects at approximately 8.3, 8.55 and 8.8 seconds and the lower thorax at 8.15, 8.42, 8.68, 8.94 seconds. Those values match those observed on FIGURE 3.14.

24 Hz noise It was observed in the T1 intervertebral forces in FIGURE 3.13 that a frequency of approximately 24 Hz was present between 11 and 13 seconds. This noise was found again in the output of the spline function and its derivatives. The frequency can be computed from FIGURE 3.16b:

$$f_1 = \frac{6}{12.65 - 12.41} = 25 [Hz] \approx 24 [Hz]$$

500 Hz noise This frequency was indeed, like hypothesized, tied to the muscles as can be seen in FIGURE 3.17. It also shows that the majority of the forces are present in the rear muscles. It is normal since the movement is a forward flexion of the thorax, the front muscles therefore are rarely under tension and since the muscles only work in tension, the forces inside the front muscles are equal to zero.

$$f_1 = \frac{7}{9.386 - 9.372} = 500 [Hz]$$

An important feature of the muscles forces can be observed. Indeed, 500 Hz is half the frequency of the time step of the integrator. The fact that the muscles forces are once equal to zero and the next time step equal to a value could be due to the fact that the spine oscillates between flexion and compression. More researches could be done on this feature. An improvement could be to take the envelope of the muscle forces signal.

The two major further improvements need to be done on the spline function and the `user_LinkForces` function dealing with muscles.

No problem was found in the linear interpolation function. Moreover, the data coming out of this function is not derived twice so even if there was a microscopic problem, it would not be enhanced.

It is important to note that at first, all the interpolations, including those of the accelerations and the muscles lengths, were realized with the spline function. To reduce the noise, those two interpolations were realized with a linear interpolation. Moreover, a Butterworth low-pass filter (sampling frequency = 1 kHz, cutoff frequency = 5 Hz), was added between simulation 3 and 4 to remove the noise in the muscles lengths.

Ultimately, for visualization purposes, the final results of the intervertebral disk efforts (as in FIGURE 3.13) were passed through a Butterworth low-pass filter (sampling frequency = 1 kHz, cutoff frequency = 5 Hz) leading to graphs shown in SECTION 3.2.

3.5 Rigid thorax hypothesis validation and intervertebral efforts of daily life movements

3.5.1 Introduction

In order to recover the intervertebral efforts in the spine for different daily life movements, it is important at first to affirm or invalidate the hypothesis that was stated in CHAPTER 2 where the thoracic cage was assumed to be rigid.

The goal of this experiment is to analyze the intervertebral efforts of different daily life movements like kneeling, jumping, walking or bending and compare them in order to assess the most demanding movements. Indeed, people suffering from spine pathologies like lower back pain or scoliosis would suffer from high intervertebral efforts.

3.5.2 Materials and methods

Materials

- 3 x-IMU [46]
- 3 velcro bands with a pocket sewed in each one to insert the IMU in it (FIGURE 3.18)
- A structure to tare the IMUs in it (FIGURE 3.19)
- A device to release a magnetic discharge peak (FIGURE 3.20)
- 1 computer with a micro-sd card reader or an additional reader plugged-in via usb
- 3 micro-sd cards
- x-IMU GUI software [46]
- A test subject (height = 1m60)



Figure 3.18: Velcro band with x-IMU in it



Figure 3.19: Structure to tare the IMUs

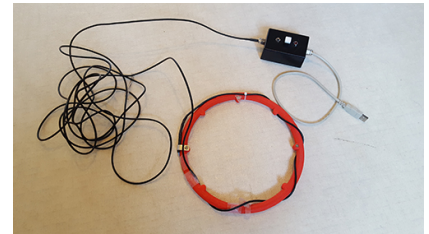


Figure 3.20: Magnetic device

Methods

First part : data acquisition

The experiment began with switching the IMUs on via the on/off switch (no micro-sd card in the IMUs yet). After that, they were connected via Bluetooth to the software[46]. The registers were written (acquisition frequency (128Hz), mode (IMU), micro-sd file number (00000.bin), side button mode (sleep/wake)) and the IMUs were tared on the same rack (FIGURE 3.19), exactly like shown in FIGURE A.1, the test subject facing the X-axis.

Due to problems when writing on the sd card in AHRS mode (no problems occurred in the Bluetooth mode (see SECTION 3.1 for the bluetooth experiment)), the IMU mode was chosen since there was not

a big difference in the drift (TABLE 3.1). The only condition was to not make the movements too fast. Indeed, in an experiment in [33], it was found that the speed had an impact on the precision of the sensors. It was said that the x-IMU had to converge towards the right solution and that it could take a certain amount of time to obtain the correct orientation. The AHRS mode deals more easily with this problem because it uses Earth's magnetic field to recalibrate itself, in addition to the gravity.

The micro-sd card works this way : a binary file is created each time the x-IMU is switched on, reset or wakes up. Logging stops once the x-IMU is reset or enters sleep mode. Each time a binary file is closed, the file name is incremented. You have to close the binary file to be able to read it or the data will be corrupted.

Once the calibration was achieved, the side button was pushed to enter sleep mode. Then, the micro-sd cards were inserted in the IMUs. Thanks to velcro in the pockets and on the IMUs, the IMUs were fixed on the bands who were attached on the test subject. One band was positioned on the pelvis and the two others on the thoracic cage, one as low as possible and the other one as high as possible (FIGURE 3.21).

As soon as the IMU's were fixed on the person, the side button was pushed, the IMUs entering wake mode and starting to log data on a binary file. Then, the magnetic device (FIGURE 3.20) was linked to the computer via a usb plug, powering it up. The circle was approached near the subject's back (between 5 and 20cm) and the button on the magnetic device was pushed. The movement could be recorded right after. To finish the logging of the data, the side button on the IMUs was pushed. An additional movement could be recorded by pushing the side button again, pushing the button of the magnetic device, doing the movement, and then pushing back the side button on the IMUs.

5 movements were made this way : flexion, lateral right and left flexion, and right and left rotation of the torso. After that, the IMUs were removed from the person and then tared again on the rack to avoid a possible drift (around $1.5^\circ/\text{min}$ (TABLE 3.1 and FIGURE 3.5)). 5 more movements were made on the second batch: put the knees on the ground, random movement of the trunk, two walks (around 3 steps) and the last file contained two jumps.



Figure 3.21: Positioning of the IMUs

Second part : post-processing, spine kinematics and dynamics

This second part was explained fully in SECTION 2.4. The frames used in this experiment are shown in FIGURE 2.5.

3.5.3 Results and discussion

Rigid thorax hypothesis

For the validation of the rigid thorax hypothesis, the kinematics and dynamics of the S1-L5 intervertebral disk for a forward flexion, two jumps and a three steps walk, were studied. The other movements and intervertebral disks were behaving the same way.

Results

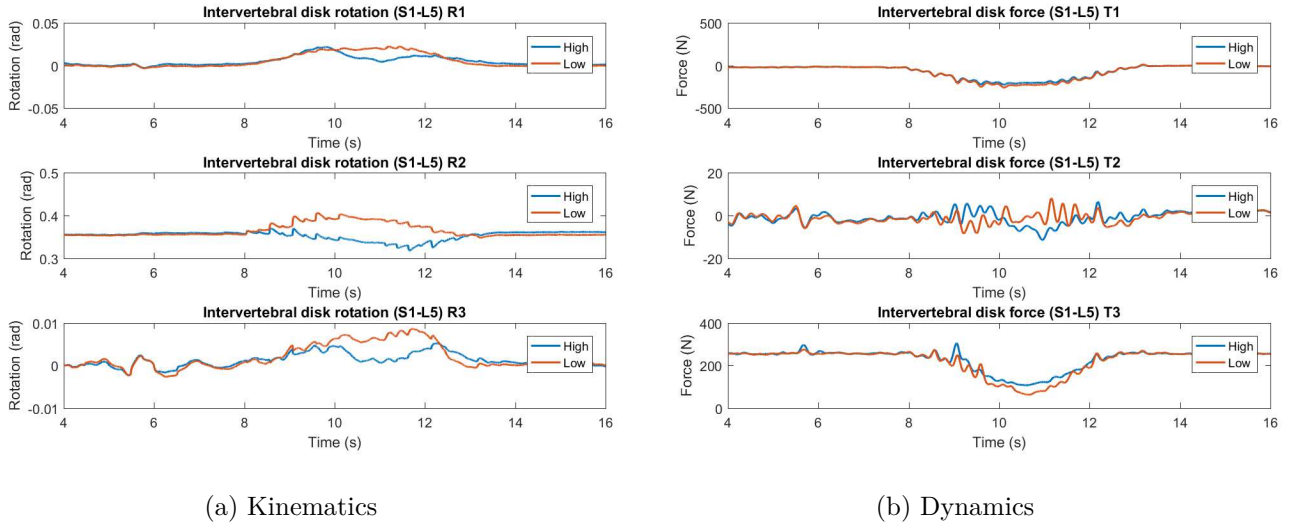


Figure 3.22: Kinematics and dynamics of the S1-L5 intervertebral disk for a forward flexion

The error percentage was computed for the intervertebral disks efforts between 8 and 13 seconds (duration of the movement).

	T1 [%]	T2 [%]	T3 [%]
Error rate	5.2	too much noise	9.9

Table 3.4: Error rate for the flexion movement

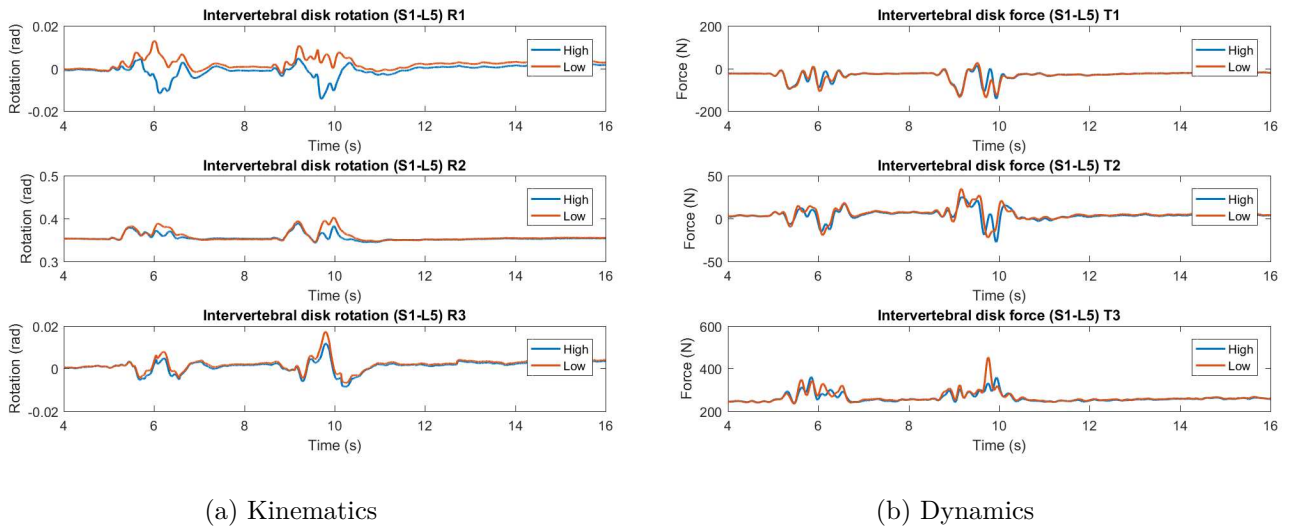


Figure 3.23: Kinematics and dynamics of the S1-L5 intervertebral disk for 2 jumps

The error percentage was computed for the intervertebral disks efforts between 5 and 7 seconds (1st jump) and 9 and 11 seconds (2d jump).

	T1 [%]	T2 [%]	T3 [%]
Error rate (1st)	6.44	7.78	12.95
Error rate (2d)	9.78	11.23	12.78

Table 3.5: Error rate for the two jumps

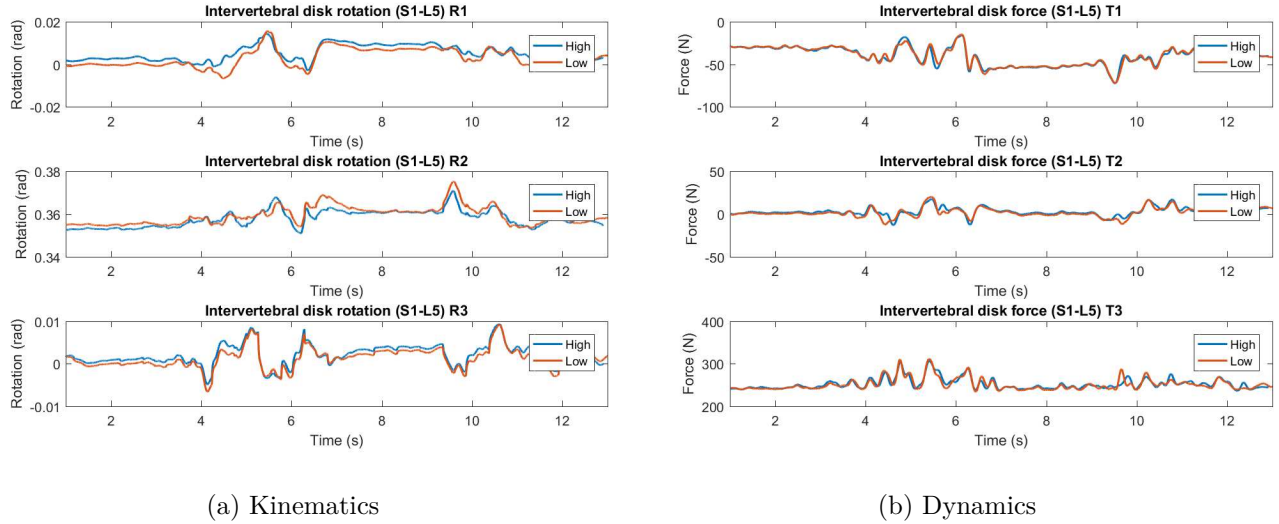


Figure 3.24: Kinematics and dynamics of the S1-L5 intervertebral disk for 3 steps (walk)

The error percentage was computed for the intervertebral disks efforts between 3 and 12 seconds (duration of the movement).

	T1 [%]	T2 [%]	T3 [%]
Error rate	2.6	6.9	6.6

Table 3.6: Error rate for the walking movement

Discussion

The error of the two jumps are higher than the other two movements. As stated in the previous thesis [33], the IMUs have more problems with higher speeds. It can be seen that the walking movement does not have a high error rate. It can be explained by the fact that the spine is bending less when walking than jumping and performing a forward flexion.

It can be seen in FIGURES 3.22, 3.23, and 3.24 that although the kinematics of the intervertebral disk S1-L5 are not always exactly the same, the dynamics are similar. Therefore, the hypothesis can be validated.

Intervertebral efforts of daily life movements

The maximal intervertebral (S1-L5) forces of daily life movements were studied with respect to the values obtained for the test subject at rest, standing.

Movement		T1 [N]	T2 [N]	T3 [N]
At rest (standing)		-29	0	242
Forward flexion	Max	-6.74	8.11	258.15
	Min	-250.24	-8.42	63.87
Lateral left/right flexion	Max	-6.08/-4.11	13.58/147.6	307.95/295.7
	Min	-63.22/-76.37	-169.8/-1.75	181.28/197.17
Left/right rotation (trunk)	Max	-21.71/7.88	30.22/19.55	329.7/383.72
	Min	-82.29/-39.09	-21.26/-27.41	242.8/263.95
Knees on the floor	Max	61.33	89.25	1021.35
	Min	-131.4	-71.64	233.67
Random movement (trunk)	Max	-31.93	18.76	799.36
	Min	-206.42	-26.57	153.8
Walk	Max	-15.15	20.07	311.41
	Min	-71.68	-8.41	234.39
Jump	Max	26.45	34.72	452.16
	Min	-133.54	-22.15	237.65

Table 3.7: Minimal and maximal values for the intervertebral (S1-L5) efforts of different daily life movements

At rest (standing) The spine supporting the mass of the upper body, it is therefore normal that the majority of intervertebral efforts when standing are in the T3 direction. The efforts in the T2 direction are close to zero since the spine of a healthy subject is symmetric. A scoliotic patient would have values different from zero at rest. The efforts along T1 are slightly negative because the L5 vertebra is tilting forward.

Forward flexion As already explained in SECTION 3.3, it is normal that the effort value in T3 diminishes during the movement with respect to the rest position. Efforts along T1 are also reduced since when the trunk moves forward, the force in the intervertebral disk points backwards. There is almost no change along T2 since there is no lateral movement. This movement can be seen in FIGURE 3.22b.

Lateral left/right flexion The values in T1 in FIGURE 3.25 indicate a small forward flexion during the left and right lateral flexion. For the T2 values, a symmetry can clearly be seen. Indeed, for a lateral flexion to the left, a negative value can be observed since the force points towards the right. The inverse can be observed for a lateral flexion to the right. For the T3 direction, the efforts are reduced during the movement with respect to the standing position. Indeed, the same thing happened with the forward flexion of the trunk and is explained in SECTION 3.3. The only difference with the forward flexion is that the couple T2/T3 is concerned here and not the T1/T3 couple. In reality, the maximal value of T3 is closer to 285N than the value stated in TABLE 3.7 because some important peaks of noise can be observed. Therefore, the delta efforts along T3 for the lateral flexion are equal to $\approx 40\text{N}$, which is way

smaller than the efforts along T3 for the forward flexion ($\approx 245\text{N}$). Indeed, the human body has less flexibility during a lateral flexion than a forward flexion, therefore, the test subject had a biggest tilt angle during the forward flexion. Moreover, the delta of the efforts in T2 (direction of flexion) for the lateral flexion ($\approx 170\text{N}$) is smaller than the delta of the efforts in T1 (direction of flexion) for the forward flexion ($\approx 245\text{N}$), which supports the argument.

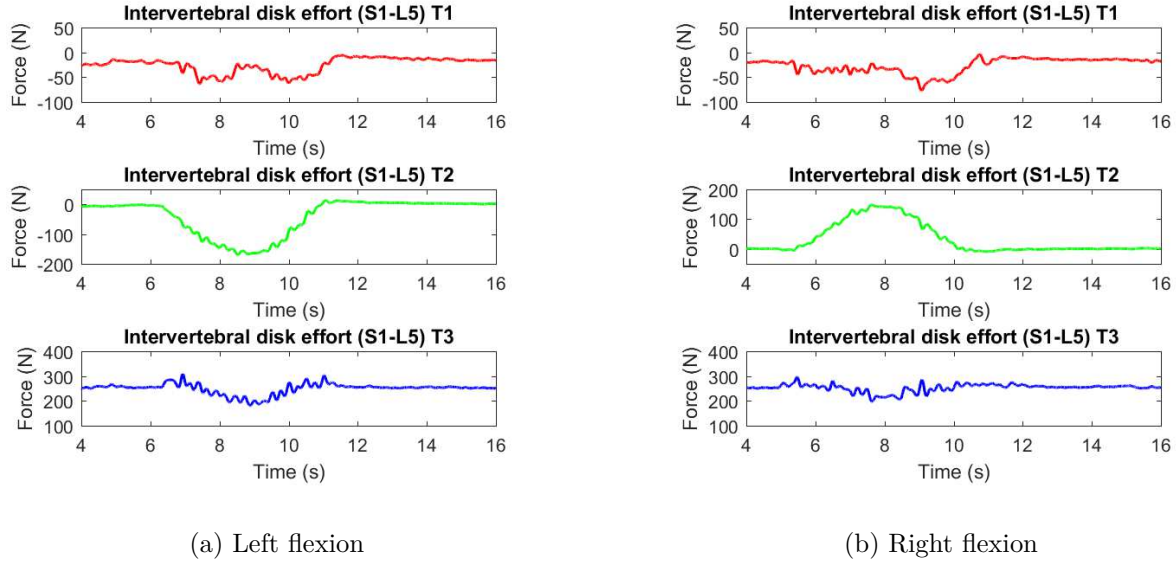


Figure 3.25: Dynamics of the S1-L5 intervertebral disk for left and right flexions of the trunk

Left/right rotation (trunk) There is no significant change of efforts along the T1 and T2 directions in FIGURE 3.26, but a small delta effort ($\approx 60\text{N}$) can be observed in T1 for the left rotation, the efforts being reduced during the movement indicating a slight forward tilting of the test subject. For the right rotation, the efforts are increased ($\approx 47\text{N}$) during the movement, indicating a small backwards tilting. A symmetry can also be observed for the T2 direction. Indeed, for the left rotation movement, the efforts are first positive then negative. It is normal since when the subject turned its trunk to the left, their right shoulder lowered down, the spine tilting towards the right, a positive effort along T2 therefore being observed. The opposite can be observed for the right rotation movement. There is no clear shape for the T3 efforts curve, since there is only peaks of force higher than the rest position.

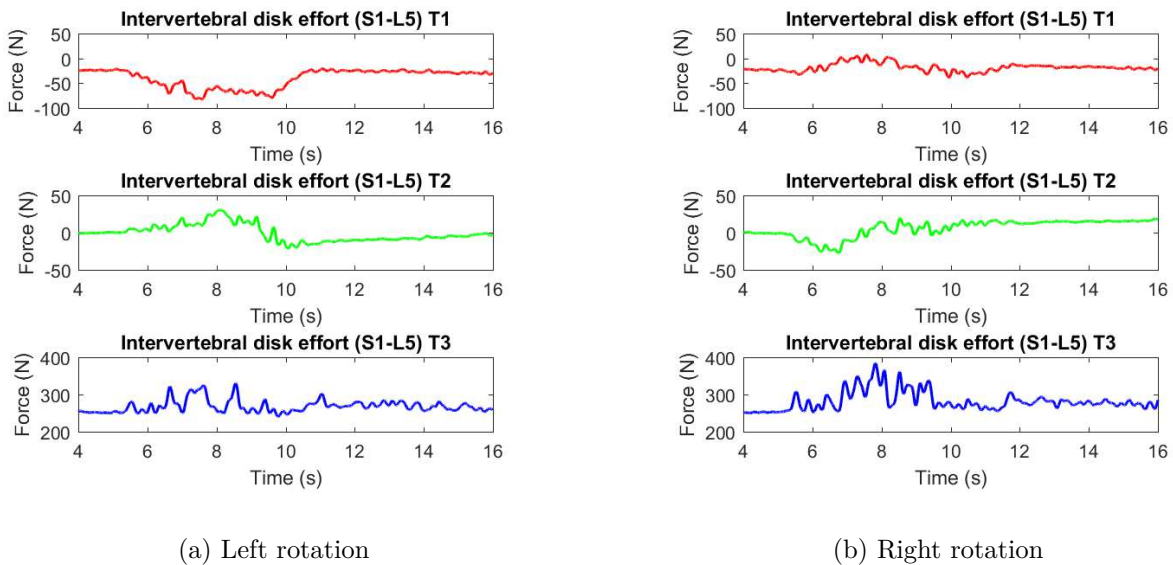


Figure 3.26: Dynamics of the S1-L5 intervertebral disk for lateral left and right rotations of the trunk

Knees on the floor First, the test subject put the left knee on the floor, keeping a straight back. Therefore, in FIGURE 3.27, the efforts along the T1 direction diminish because of a forward tilt of the trunk. Then, when the right knee is laid on the floor, the spine tilts towards the left, inducing negative efforts in T2 and an important peak in T3 (812.6 N). After that, the test subject sits on their heels, inducing an important increase in T1 efforts ($\approx 180\text{N}$). T2 stays around zero since the spine is straight. When the left knee is raised, the spine tilts to the right, inducing positive T2 efforts. Moreover, it can be observed that this is a symmetric movement, especially in T2 since the absolute values of the two peaks are almost equal ($89.25\text{ N} \approx |-71.64|\text{ N}$). A second peak in the T3 efforts (1021.35 N) can be observed when the test subject recovers on their feet, because a high vertical acceleration occurs at that time. This is a demanding movement in terms of vertical efforts since the efforts along T3 are the highest of TABLE 3.7.

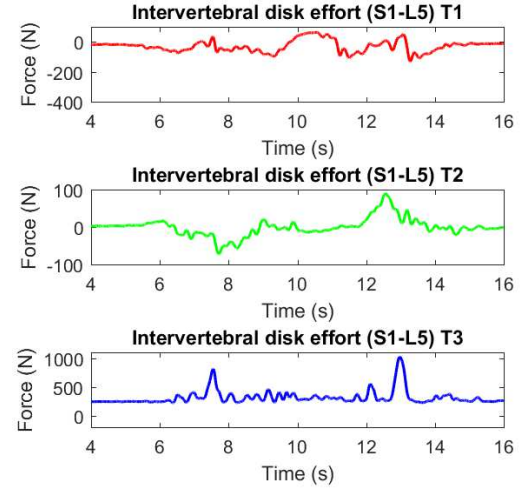


Figure 3.27: Efforts in the S1-L5 intervertebral disk for the kneeling movement

Random movement of the trunk The random movement was a forward tilt, stopped when the spine was bent, then followed by a turning motion to the left, then a recovery to the standing position. It gives high values in T3 (799.36 N) because the tilting of the spine was quite pronounced during the movement. That is also why T1 diminishes, too.

Walk The main efforts are located along the T3 direction which is normal since the intervertebral disk had to support the entirety of the upper body and additional vertical shocks during the walk. That is why all the efforts are higher than the resting position. The movement can be seen in FIGURE 3.24.

Jump Like the walking movement, the jumping movement mainly consists of T3 efforts, which are higher than the resting position, too. Moreover, the T3 efforts are higher in the jumps (452.16 N) than when walking (311.41 N) because the shocks are higher when jumping. The T1 efforts are diminishing during the movement since the spine is tilting forward during the jump. The movement can be seen in FIGURE 3.23.

In conclusion, the efforts are larger when shocks and important accelerations are occurring. Movements like kneeling or jumping induce important efforts that may cause problems in the intervertebral disks in the long run. Moreover, G. Abedrabbo [8] says that for scoliosis, *“in addition to the increased intervertebral forces, there is a significant change in the way forces are distributed along the spine. This means that a simplified model does not only underestimate those forces, but also shows a skewed behavior compared to a physiological model”*. Indeed, those important efforts during daily life movements could cause even more problems for people with spine pathologies like scoliosis.

The main goal of this master thesis was to simplify the protocol for data acquisition in order to obtain the intervertebral efforts in the lumbar spine. To this end, optoelectric motion systems were replaced by Inertial Measurement Units, more affordable and portable sensors capable of recording the acceleration, angular speed, and magnetic field in order to compute the Euler angles of the sensor with respect to its inertial frame. To this end, a multibody model of the spine was created in order to obtain the kinematics of the lumbar spine thanks to the Euler angles obtained with the sensors. Steps to obtain the kinematics and dynamics of the spine were also developed as well as a computer implementation in Matlab and C languages.

A few experiments were conducted in order to answer the questions asked at the beginning of this master thesis. The experiments allowed us to observe that internal spinal efforts depends heavily on the shocks and accelerations as seen in the kneeling and jumping movements. The x-IMUs giving excellent accelerations [33], they are perfect to analyze the intervertebral efforts. It was also seen that a change of inertia of the lumbar vertebrae were having an effect on the efforts, too much inertia skewing the results. It was also found that the muscles were adding efforts in the vertebrae since they pulled on the spine in order to produce movements. A model without muscles therefore underestimates the forces at play. Moreover, this model was found to be quite sensitive to noises coming from the interpolation function and the muscles. It also allowed us to find that thanks to the validated rigid thorax hypothesis, two IMUs were sufficient in order to correctly assess intervertebral efforts.

This model already allows us to correctly assess intervertebral efforts and analyze the differences in different daily life movements for the lumbar region, but it still needs to grow. The perspective would first be to verify the values obtained for the daily life movements with literature data for healthy subjects. The literature on the subject was not found, because not easy to find, for this master thesis but an in-depth research on the matter needs to be realized. In order to be clinically usable (e.g. for the surgical planning of scoliotic patients), a number of improvements and researches could be achieved :

- Each person is different, with various spine geometries and pathologies. That is why this patient-specific models needs to be improved. The model should be calibrated exactly to the person doing the measures (which was not the case ins this master thesis). In order to be as close as possible to the reality, measurement of segment length should be performed on the person as well as their weight taken. The majority of the inertia and masses are already added via the C program and can easily be changed, but the inertia and weight of the upper body are computed thanks to an external Matlab program. It should therefore be translated into the C language and embedded in the program. Moreover, the inertia computed in order to find the anthropometric data were too small and had to be increased by factors of 10 and 50. The improved model should be able to work with small inertia as close as possible to the real values.
- In order to get better time performances, the entirety of the program should be written in C, including the data processing.
- The combination of the spine and the muscles maintains the person standing. The benefit of added muscles in the model was already shown. But this model only represents 4 muscles, the human body containing a lot more. Therefore, more muscles should be added to this model. Moreover, it was observed that the modeled muscles induced noise in the final results. An amelioration of this model should be performed, maybe in taking the envelope of the muscles signal. Additionally, a parametric analysis could be performed on the effect of muscles on the intervertebral efforts of the lumbar spine.

- It was seen in the introduction that skin movements occurred, unavoidably creating errors between the coordinates of the vertebrae and those of the sensors. The skin movements should therefore be analyzed and taken into account if necessary.
- The translations of the spine are blocked in this model (ideal articulations), but in reality, there is a slight friction in translation in the intervertebral disks. It should therefore also be analyzed with respect to intervertebral efforts and taken into account if necessary.
- Perform a parametric analysis on the gains of the different algorithms used in order to get the Euler angles. It was effectively seen that the Magdwick algorithms had good results in terms of delay. A code should be written in order to use the external algorithms if needed. The magnetometer could therefore be used in the computation of Euler angles, reducing the drift in the yaw angle, and this change could therefore remove the problems which occurred with the μ SD cards and their frames since it would only be based on the raw data.

“All models are wrong, but some are useful” is a well known quote by George Box [62]. Indeed, it will never be possible to know one hundred percent the reality thanks to models, but we can head towards it by providing patient-specific data to those models. Moreover, they allow us to understand the underlying mechanisms behind spine mechanics and related pathologies.

A project cannot function without proper organization, problems solving, and time management. A balance needs to be found between meetings with the supervisor, writing the report, performing researches, experiments and modeling (FIGURES A.3, A.4, and A.5). Moreover, performing experiments was hindered by the fact that the IMUs were not always available to use. In the first semester I also had a lot of different other courses with projects that needed to be realized as well. These constraints were reduced in the second semester since the number of courses was lower than in the first semester. An important part of the project management was to learn how to handle unexpected events.

I first started to do researches on the spine (geometry, anatomy, ...) and the IMUs (how to use them, their different algorithms, ...). But soon, the modeling part had to start.

The first model (FIGURE A.3) was coded in Matlab. It is a simple body with 3 degrees of freedom controlled via the 3 Euler angles of the real IMU (roll (R1), pitch (R2), yaw (R3)). The goal of this model was to understand how to use the IMU data as an input of the model.

At the start of the second semester, I was asked to translate my Matlab code to the C language, first of all for a question of speed but also to use the same language as G. Abedrabbo to help him in his work. The problem of the C language compared to Matlab is the lack of high-level built-in functions. Therefore, a lot of functions like the spline one where it was just a call to it in Matlab had to be coded from scratch or searched on the internet.

The second model (FIGURE A.4) (coded in C) was built to test the link between an IMU and a vertebra. This model was more detailed than the first model since it contained a body for the IMU but also a body for the vertebra. Translations were also added to the loop formed by the two bodies.

Then, the second model has been duplicated to test the system with two IMUs. A big problem occurred at that moment; I was using the Bluetooth mode of the IMUs to capture the data of interest but the bluetooth application failed when using it with two IMUs. Just before performing the final experiment, I had to understand how to use the IMUs with the μ SD cards. A few Matlab programs had to be made to synchronize the files since the operational mode between μ SD cards and bluetooth is not the same. In addition to that, other problems emerged; The frame of the IMUs in the μ SD mode was different than in the bluetooth mode; The AHRS mode was experiencing problems when using the IMUs with the μ SD cards. Moreover, I only had the μ SD cards for four days. Waiting for the μ SD cards allowed me to already review the spine model of G. Abedrabbo and add the IMUs to it. Before the change from Bluetooth to μ SD cards, the entirety of the Euler angles transformations and initial data processing were realized in the C code and not via Matlab, therefore, the only program needed was Visual Studio.

A small experiment with a mechanical device made in K'NEX (FIGURE A.1) has been realized to validate the second model (sequence of the Euler angles) by turning the IMU around the X, Y and Z axis. The calibration and frames of the IMUs (IMU (left) and IMU.2 (right)) and the layout of the experiment are showed in FIGURES A.1 and A.2 respectively.

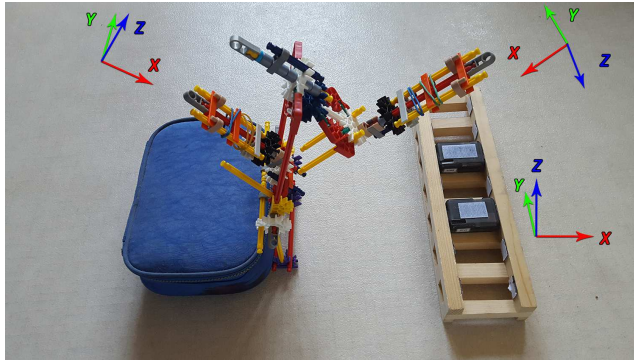


Figure A.1: Calibration axis

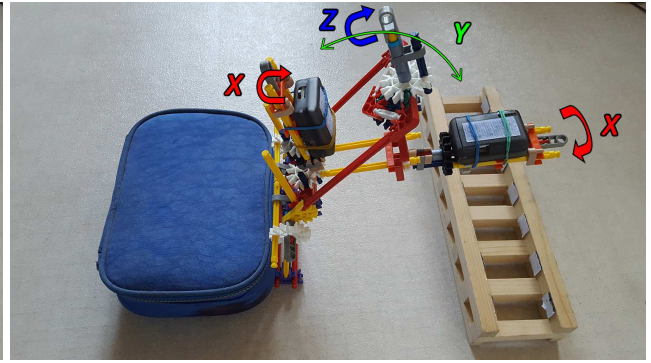


Figure A.2: Axis of rotation

The experiment was filmed and the video was compared to the animation given by Robotran. Thanks to that comparison, the model was validated.

Soon, G. Abedrabbo and I saw that the model of the spine could be enhanced to increase the level of comprehension of people external to the modeling process. That is how the third model was born (see CHAPTER 2).

At the end of week eleven, it was decided to add muscles to the model and to analyze the intervertebral efforts of the spine. The addition of the muscles to the third model is called the fourth model in FIGURE A.5.

The analysis of the kinematics and dynamics of the spine and the completion the report were realized in the last few weeks. At that moment, I had to juggle between that and my two first exams.



1st semester

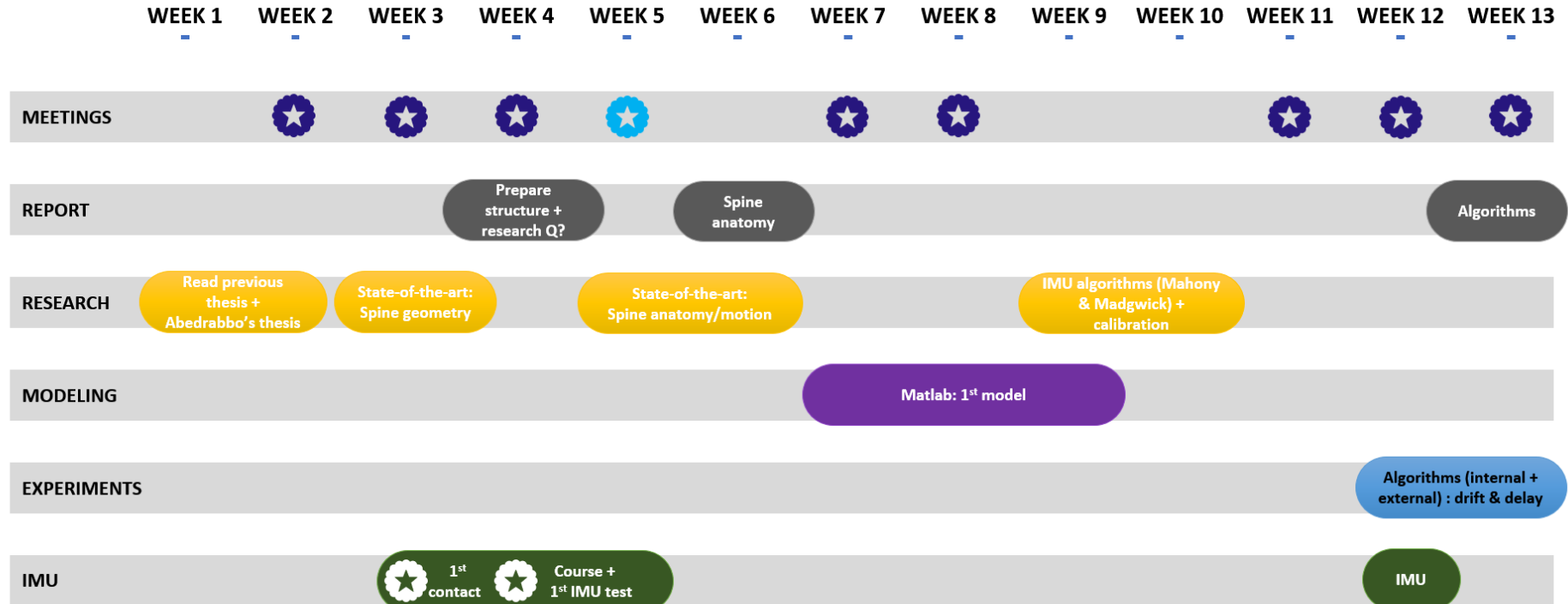


Figure A.3: 1st semester organization



2^d semester (1)

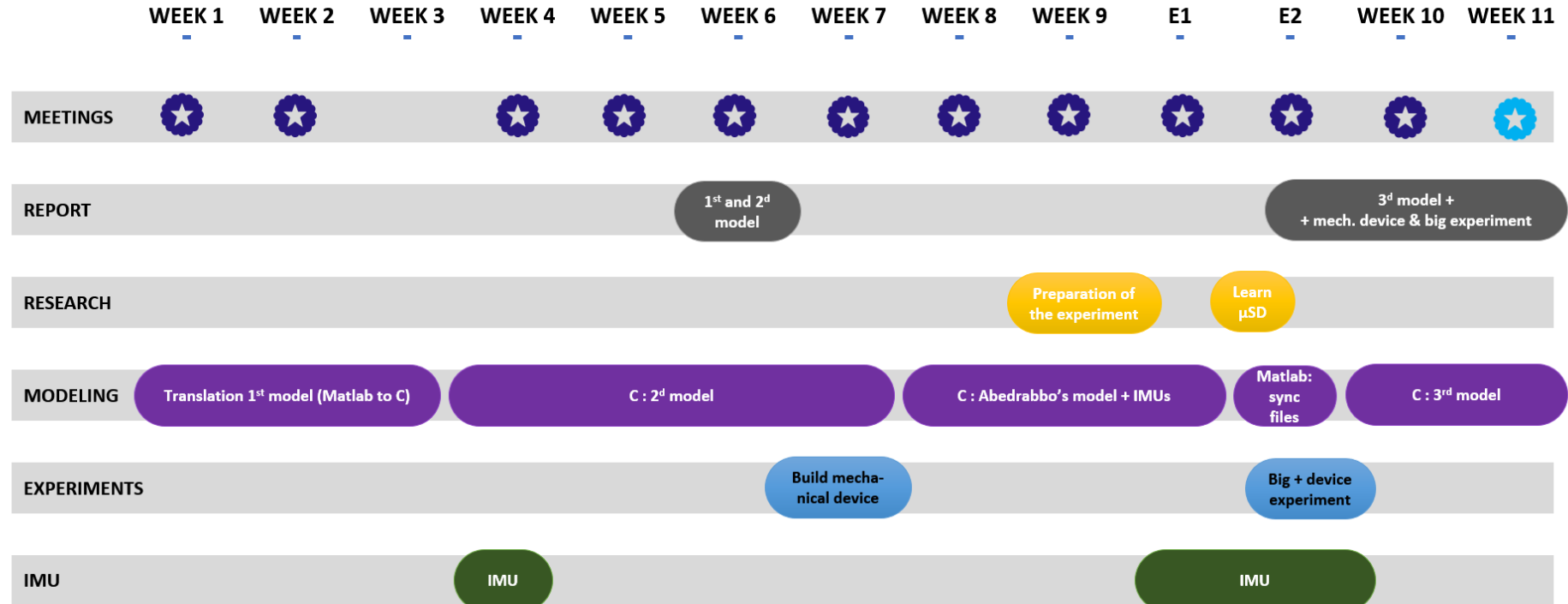


Figure A.4: 2d semester organization: part 1



2^d semester (2)

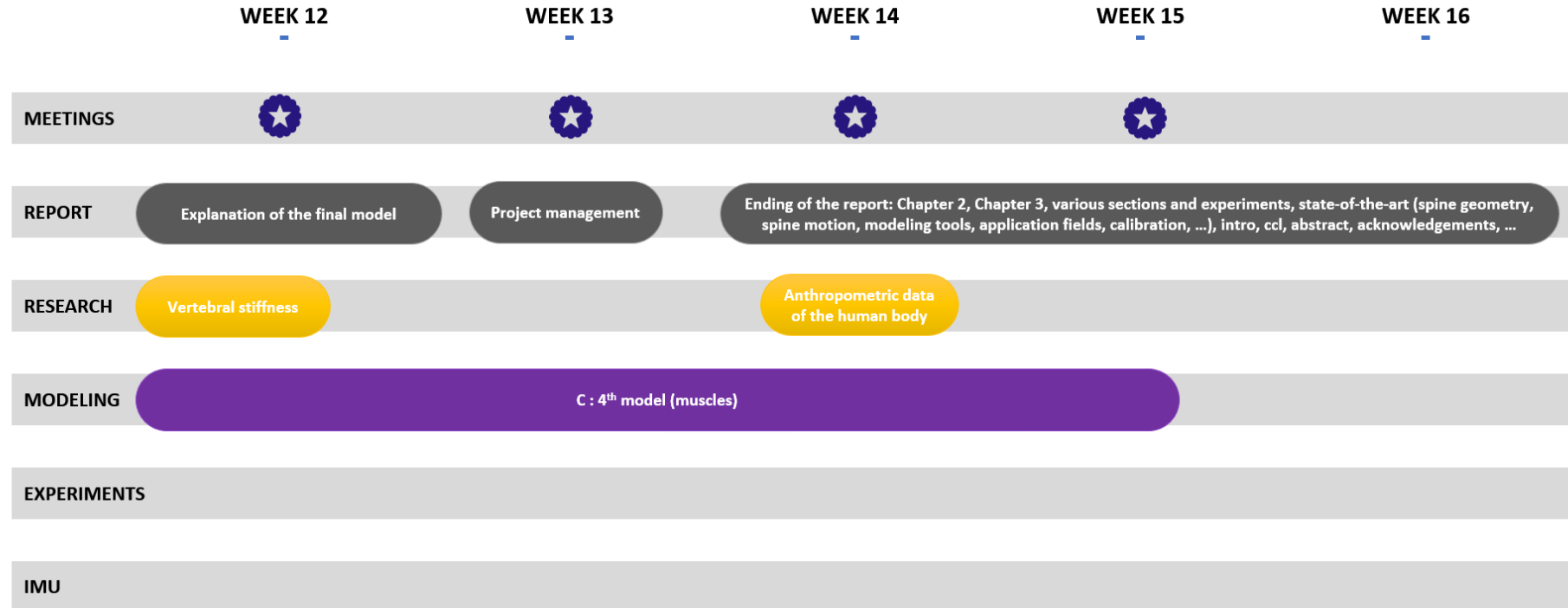


Figure A.5: 2d semester organization: part 2

B.1 Matlab code: synchronization

```

close all;
clear all;

pelvis = 1;
5
%% Open Acc, Gyr, Magn

Creation_mat_file('Sauts_Pelvis_CallInertialAndMag.csv');

10 load('Donnees_brutes.mat');

%% Open Euler

15 eulerfichier=csvread('Sauts_Pelvis_EulerAngles.csv',1,0);

packet = eulerfichier(:,1);
roll = deg2rad(eulerfichier(:,2));
pitch = deg2rad(eulerfichier(:,3));
20 yaw = deg2rad(eulerfichier(:,4));

fq = 128;
time = packet/fq;

25 %% Filter data

for i = 1:3
    Accelerometer_filt(:,i) = filtrage(Accelerometer(:,i), 128, 5);
end

30 %% Seeing where to cut thanks to the magnetometer

max_vector = [max(abs(Magnetometer([5:end],1))) max(abs(Magnetometer([5:end],2)
    ↪ )) max(abs(Magnetometer([5:end],3)))];

35 if (max_vector(1) > max_vector(2) && max_vector(1) > max_vector(3))
    MAX_magn_axis = 1;
elseif (max_vector(2) > max_vector(1) && max_vector(2) > max_vector(3))
    MAX_magn_axis = 2;
else
40     MAX_magn_axis = 3;
end

MAX_magn = max(max_vector);

45
if (MAX_magn_axis == 1)
    index_max = find(abs(Magnetometer([5:end],1))==MAX_magn);
elseif (MAX_magn_axis == 2)

```

```

        index_max = find(abs(Magnetometer([5:end],2))==MAX_magn);
50 else
        index_max = find(abs(Magnetometer([5:end],3))==MAX_magn);
    end

    index_max_value = index_max(1)+30; % go further than the peak
55
    %% Cut the Euler angles file

    cut_roll = roll(index_max_value:end);
    cut_pitch = pitch(index_max_value:end);
60 cut_yaw = yaw(index_max_value:end);
    cut_time = time(index_max_value:end);

    for i = 1:3
65 cut_Accelerometer_filt(:,i) = Accelerometer_filt(index_max_value:end,i);
    end

    %% Start the time at 0

70 final_time = cut_time-cut_time(1);

    %% Transform the data

    unwrap_roll = filtrage(unwrap_angle(cut_roll, (3*pi/4)),128,5);
75 unwrap_pitch = filtrage(unwrap_angle(cut_pitch, (3*pi/4)),128,5);
    unwrap_yaw = filtrage(unwrap_angle(cut_yaw, (3*pi/4)),128,5);

    figure
80 plot(final_time, unwrap_roll*(180/pi), final_time, unwrap_pitch*(180/pi),
        ↪ final_time, unwrap_yaw*(180/pi), 'LineWidth', 1);
    title('Euler angles UNWRAPPED');
    legend('Roll','Pitch','Yaw','Location','NorthWest');
    xlabel('Time (s)');
    ylabel('Angle (degrees)');
85 grid on;
    savefig('Sauts_Pelvis.fig')

    %% Write the csv file
90
    Matrix = [unwrap_roll unwrap_pitch unwrap_yaw final_time];

    if pelvis == 1
        csvwrite('D:\_Documents\Unif\Master thesis\X-IO\Donnees\Mardi essai sd
            ↪ colonne\Final files\Flexion_Pelvis_matrix.csv', Matrix)
95 csvwrite('D:\_Documents\Unif\Master thesis\X-IO\Donnees\Mardi essai sd
            ↪ colonne\Final files\Flexion_acc.csv', cut_Accelerometer_filt)
    else
        csvwrite('D:\_Documents\Unif\Master thesis\X-IO\Donnees\Mardi essai sd
            ↪ colonne\Final files\Flexion_Th_bas_matrix.csv', Matrix)
    end

```

B.2 Matlab code: inertia calculation

```

%%
% Find the moments of inertia about the center of mass of the system
clc;
clear all;
5 % head | trunk | right upper arm | left upper arm | right forearm | left
  ↳ forearm
m_f = 56; % [kg]
M = [6.68, 15.45, 2.55, 2.55, 1.38, 1.38]*m_f/100; % [kg]
L = [200.2, 142.5, 275.1, 275.1, 264.3, 264.3]/1000; % [m]
L_cg = [58.94, 20.77, 57.54, 57.54, 45.59, 45.59].*L/100; % [m] par rapport au
  ↳ dessus
10 % Rx = sagittal
Rx = [33.0, 74.6, 27.8, 27.8, 26.1, 26.1].*L/100; % [m]
% Ry = transverse
Ry = [35.9, 50.2, 26.0, 26.0, 25.7, 25.7].*L/100; % [m]
15 % Rz = longitudinal
Rz = [31.8, 71.8, 14.8, 14.8, 9.4, 9.4].*L/100; % [m]

% wrt T12
X = [0, 0, 0, 0, 0, 0];
20 Y = [0, 0, -0.18, 0.18, -0.18, 0.18];
Z = [L(2)+L(1)-L_cg(1), L(2)-L_cg(2), L(2)-L_cg(3), L(2)-L_cg(4), L(2)-L(3)-
  ↳ L_cg(5), L(2)-L(4)-L_cg(6)]; % [m]

mt = sum(M); % The total mass of this system
%%
25 % Three components of the position vector of the center of mass are
Xcg = (1/mt)*sum(M.*X);
Ycg = (1/mt)*sum(M.*Y);
Zcg = (1/mt)*sum(M.*Z);
V_cg = [Xcg, Ycg, Zcg] % [m]
30 %%
Ig = [ 0.0, 0.0, 0.0;
      0.0, 0.0, 0.0;
      0.0, 0.0, 0.0];
% head | trunk | right upper arm | left upper arm | right forearm | left
  ↳ forearm
35 Ip = [ 0.0, 0.0, 0.0, 0.0, 0.0, 0.0; % Ixx
        0.0, 0.0, 0.0, 0.0, 0.0, 0.0; % Iyy
        0.0, 0.0, 0.0, 0.0, 0.0, 0.0]; % Izz
%%
% The total moment of inertia is the sum of moments of inertia for all
40 % point masses in the system
for i = 1:6
    x = (X(i) - Xcg);
    y = (Y(i) - Ycg);
    z = (Z(i) - Zcg);
45 m = M(i);

    Ig = m*[ Rx(i)^2, 0, 0;
             0, Ry(i)^2, 0;
             0, 0, Rz(i)^2];

50 Ip_bis = Ig + [ m*(y^2 + z^2), -m*x*y, -m*x*z;
                  -m*y*x, m*(x^2 + z^2), -m*y*z;
                  -m*z*x, -m*z*y, m*(x^2 + y^2)];

```

```

                                -m*x*z,          -m*y*z,          m*(y^2 + x^2)]; % [kg*m
                                ↪ ^2]

55     Ip(:,i) = [      Ip_bis(1,1);
                    Ip_bis(2,2);
                    Ip_bis(3,3)]; % [kg*m^2]
end

60 Ip_sum = [      sum(Ip(1,:));
              sum(Ip(2,:));
              sum(Ip(3,:))]; % [kg*m^2]

```

B.3 C code: Butterworth implementation

```

#include <stdio.h>
#include <stdlib.h>
#include <math.h>

5 #include "butterworth.h"

#include "useful_functions.h"
#include "useful_functions_Eve.h"

10 //-----
// This function returns the data filtered. Converted to C# 2 July 2014.
// Original source written in VBA for Microsoft Excel, 2000 by Sam Van
// Wassenbergh (University of Antwerp), 6 june 2007.
// https://www.codeproject.com/Tips/1092012/A-Butterworth-Filter-in-Csharp
15 // Translated from C# to C by Eve Leroy
//-----

double* Butterworth(double* indata, double deltaTimeinsec, double CutOff)
{
    if (indata == NULL) return NULL;
    if (CutOff == 0) return indata;

20     double Samplingrate = 1 / deltaTimeinsec;
    long dF2 = real_length_vector(indata) - 1; // The data range is set with
    ↪ dF2
    double* Dat2 = get_dvec_0(dF2 + 4); // Array with 4 extra points front and
    ↪ back
25     double* data = indata; // Ptr., changes passed data

    int r;

    // Copy indata to Dat2
30     for (r = 0; r < dF2; r++) {
        Dat2[2 + r] = indata[r];
    }
    Dat2[1] = Dat2[0] = indata[0];
    Dat2[dF2 + 3] = Dat2[dF2 + 2] = indata[dF2];

35     const double pi = 3.14159265358979;
    double wc = tan(CutOff * pi / Samplingrate);
    double k1 = 1.414213562 * wc; // Sqrt(2) * wc
    double k2 = wc * wc;
40     double a = k2 / (1 + k1 + k2);
    double b = 2 * a;
    double c = a;

```

```

double k3 = b / k2;
double d = -2 * a + k3;
45 double e = 1 - (2 * a) - k3;

// RECURSIVE TRIGGERS - ENABLE filter is performed (first, last points
//   ↪ constant)
double* DatYt = get_dvec_0(dF2 + 4);
DatYt[1] = DatYt[0] = indata[0];
50
for (long s = 2; s < dF2 + 2; s++) {
    DatYt[s] = a * Dat2[s] + b * Dat2[s - 1] + c * Dat2[s - 2]
        + d * DatYt[s - 1] + e * DatYt[s - 2];
}
55 DatYt[dF2 + 3] = DatYt[dF2 + 2] = DatYt[dF2 + 1];

// FORWARD filter
double* DatZt = get_dvec_0(dF2 + 2);
DatZt[dF2] = DatYt[dF2 + 2];
60 DatZt[dF2 + 1] = DatYt[dF2 + 3];

int t;

for (t = -dF2 + 1; t <= 0; t++) {
65     DatZt[-t] = a * DatYt[-t + 2] + b * DatYt[-t + 3] + c * DatYt[-t + 4]
        + d * DatZt[-t + 1] + e * DatZt[-t + 2];
}

// Calculated points copied for return
70 int p;

for (p = 0; p < dF2; p++) {
    data[p] = DatZt[p];
}
75
return data;
}

```

B.4 C code: derspline

```

#include <stdio.h>
#include <stdlib.h>

#include "derspline.h"
5 #include "user_model.h"

#include "useful_functions.h"
#include "useful_functions_Eve.h"
#include "spline.h"
10

double* derspline(double* x, double* y, double xf, int length, MbsData *
//   ↪ mbs_data)
{
15     int i,j,finish;
    i = 0; j = 0;

    double** ycoef = get_dmat_0(length - 1, 4);

```

```

double** ydcoef = get_dmat_0(length - 1, 3);
double** yddcoef = get_dmat_0(length - 1, 2);

double xxi = 0;
double yf = 0, yfd = 0, yfdd = 0;

double* yf_vec = get_dvec_0(3);

double* xxiy = get_dvec_0(4);
double* xxiyd = get_dvec_0(3);
double* xxiydd = get_dvec_0(2);

double* yfline = get_dvec_0(4);
double* yfdline = get_dvec_0(3);
double* yfddline = get_dvec_0(2);

// Spline interpolation
spline(length, x, y, ycoef);

// Derivatives
for (i = 0; i < length-1; i++) {
    ydcoef[i][0] = 3 * ycoef[i][0];
    ydcoef[i][1] = 2 * ycoef[i][1];
    ydcoef[i][2] = ycoef[i][2];

    yddcoef[i][0] = 6 * ycoef[i][0];
    yddcoef[i][1] = 2 * ycoef[i][1];
}

// Data calculation
i = 0;
finish = 0;

while (finish == 0 && i < (length-1))
{
    if (xf <= x[i+1]) {
        xxi = xf - x[i];
        finish = 1;
    }
    else
    {
        i++;
    }
}

xxiy[0] = pow(xxi, 3);
xxiy[1] = pow(xxi, 2);
xxiy[2] = xxi;
xxiy[3] = 1;

xxiyd[0] = xxiy[1];
xxiyd[1] = xxiy[2];
xxiyd[2] = xxiy[3];

xxiydd[0] = xxiy[2];
xxiydd[1] = xxiy[3];

if (i != 0) {

```

```

        i--;
    }

80    getLine(i, ycoef, yflines, 4);
    getLine(i, ydcoef, yfdlines, 3);
    getLine(i, yddcoef, yfddlines, 2);

85    yf = scalar_product_0(yflines, xxiy, 4);
    yfd = scalar_product_0(yfdlines, xxiyd, 3);
    yfdd = scalar_product_0(yfddlines, xxiydd, 2);

    yf_vec[0] = yf;
90    yf_vec[1] = yfd;
    yf_vec[2] = yfdd;

    //free matrices and vectors
    free_dmat_0(ycoef);
95    free_dmat_0(ydcoef);
    free_dmat_0(yddcoef);

    free_dvec_0(xxiy);
    free_dvec_0(xxiyd);
100    free_dvec_0(xxiydd);

    free_dvec_0(yflines);
    free_dvec_0(yfdlines);
    free_dvec_0(yfddlines);

105

    return yf_vec;
}

```

B.5 C code: spline

```

/* Numerical Analysis 9th ed – Burden, Faires (Ch. 3 Natural Cubic Spline, Pg.
  ↪ 149)
https://gist.github.com/svdamani/1015c5c4b673c3297309?fbclid=
  ↪ IwAR2siaJUygmPph8RNn-A2oHK-9xw7ZakXxSOTenZ93MfPjflgPHxE0v3d0o
*/
#include <stdio.h>
5  #include "useful_functions.h"
  #include "spline.h"

void spline(int length, double* x_in, double* y_in, double** coef)
{
10    /** Step 0 */
    int n, i, j;
    n = length;
    n--;
    i = 0; j = 0;

15

    double* x = get_dvec_0(n + 1);
    double* a = get_dvec_0(n + 1);
    double* h = get_dvec_0(n);
    double* A = get_dvec_0(n);
20    double* l = get_dvec_0(n + 1);
    double* u = get_dvec_0(n + 1);
    double* z = get_dvec_0(n + 1);

```

```

double* c = get_dvec_0(n + 1);
double* b = get_dvec_0(n + 1);
double* d = get_dvec_0(n + 1);

for (i = 0; i < n + 1; i++) x[i] = x_in[i];
for (i = 0; i < n + 1; i++) a[i] = y_in[i];

/** Step 1 */
for (i = 0; i < n; i++)
    h[i] = x[i + 1] - x[i];

/** Step 2 */
for (i = 1; i < n; i++)
    A[i] = 3 * (a[i + 1] - a[i]) / h[i] - 3 * (a[i] - a[i - 1]) / h[i -
        ↪ 1];

/** Step 3 */
l[0] = 1;
u[0] = 0;
z[0] = 0;

/** Step 4 */
for (i = 1; i < n; i++) {
    l[i] = 2 * (x[i + 1] - x[i - 1]) - h[i - 1] * u[i - 1];
    u[i] = h[i] / l[i];
    z[i] = (A[i] - h[i - 1] * z[i - 1]) / l[i];
}

/** Step 5 */
l[n] = 1;
z[n] = 0;
c[n] = 0;

/** Step 6 */
for (j = n; j >= 0; j--) {
    c[j] = z[j] - u[j] * c[j + 1];
    b[j] = (a[j + 1] - a[j]) / h[j] - h[j] * (c[j + 1] + 2 * c[j]) / 3;
    d[j] = (c[j + 1] - c[j]) / (3 * h[j]);
}

/** Step 7 */
for (i = 0; i < n; i++) {
    coef[i][0] = d[i];
    coef[i][1] = c[i];
    coef[i][2] = b[i];
    coef[i][3] = a[i];
}

free(b); free(x); free(a); free(h); free(A); free(l); free(u); free(z);
    ↪ free(c); free(d);
}

```

- [1] Dictionary.com. Biomechanics. <https://www.dictionary.com/browse/biomechanics>. (visited on 26/05/19).
- [2] R. Drake, A. Wayne-Vogl and A.W. M. Mitchell. *Gray's anatomy for students*. Elsevier, third edition, 2014.
- [3] A. Middleditch and J. Oliver. *Functional anatomy of the spine*. Elsevier, second edition, 2005.
- [4] N. Bogduk. *Functional anatomy of the spine*, volume 136 of *Handbook of Clinical Neurology*, chapter 32. Elsevier, 2016.
- [5] G. Cramer, S. Darby. *Basic and Clinical Anatomy of the Spine, Spinal Cord, and ANS*. Elsevier, second edition, 2005.
- [6] R. Dumas et al. Explicit calibration method and specific device designed for stereoradiography. *Journal of Biomechanics*, page 827–834, 2003.
- [7] S. Delorme et al. Assessment of the 3-d reconstruction and high-resolution geometrical modeling of the human skeletal trunk from 2-d radiographic images. *IEEE Transactions on Biomedical Engineering*, page 989–998, 2003.
- [8] G. Abedrabbo. *Quantification of Intervertebral Efforts Using a Multibody Dynamics Approach: Application to Scoliosis*. PhD thesis, Ecole polytechnique de Louvain, Université catholique de Louvain, 2017.
- [9] V. Pomero et al. Fast accurate stereoradiographic 3d-reconstruction of the spine using a combined geometric and statistic model. *Clinical Biomechanics*, page 240–247, 2004.
- [10] E. Melhem et al. Eos biplanar x-ray imaging: concept, developments, benefits, and limitations. *Journal of children's orthopaedics*, pages 1–14, 2016.
- [11] J. Novosad et al. 3d reconstruction of the spine from uncalibrated biplanar intra-operative x-ray images. *CMBES Proceedings*, 27(1), 2002.
- [12] S. Kadoury et al. A versatile 3d reconstruction system of the spine and pelvis for clinical assessment of spinal deformities. *Medical & Biological Engineering & Computing*, page 591–602, 2007.
- [13] Wikipedia. Range of motion. https://en.wikipedia.org/wiki/Range_of_motion. (visited on 10/05/19).
- [14] W. Shiel. Range of motion. <https://www.medicinenet.com/script/main/art.asp?articlekey=5208>. (visited on 10/05/19).

- [15] W. Chiou et al. A non-invasive protocol for the determination of lumbar spine mobility. *Clinical Biomechanics*, pages 474–480, 1996.
- [16] Y.-H. Lee et al. Predictive model of intersegmental mobility of lumbar spine in the sagittal plane from skin markers. *Clinical Biomechanics*, page 413–420, 1995.
- [17] A. Rozumalski et al. The in vivo three-dimensional motion of the human lumbar spine during gait. *Gait & Posture*, page 378–384, 2008.
- [18] P. Caravaggi et al. Kinematics of the cervical spine after unilateral facet fracture. *Spine*, page E1042–E1049, 2017.
- [19] G. Mellin et al. Measurement of thoracolumbar posture and mobility with a myrin inclinometer. *Spine*, page 759–762, 1986.
- [20] M.A. Esola et al. Analysis of lumbar spine and hip motion during forward bending in subjects with and without a history of low back pain. *Spine*, page 71–78, 1996.
- [21] A. Mannion et al. A comparison of two motion analysis devices used in the measurement of lumbar spinal mobility. *Clinical Biomechanics*, page 612–619, 1999.
- [22] M. Tojima et al. Novel 3-dimensional motion analysis method for measuring the lumbar spine range of motion. *Spine*, page E1327–E1333, 2013.
- [23] M. Lee et al. A model of spine, ribcage and pelvic responses to a specific lumbar manipulative force in relaxed subjects. *Journal of Biomechanics*, 28:1403–1408, 1995.
- [24] D. Coombs et al. Simulation of lumbar spine biomechanics using abaqus. 06 2019.
- [25] 3DS. Abaqus. <https://www.3ds.com/fr/produits-et-services/simulia/produits/abaqus/>. (visited on 06/06/19).
- [26] M. Rao et al. Explicit finite element modeling of the human lumbar spine. *Electronic Theses and Dissertations*, 2012.
- [27] H. Jaramillo Suárez et al. A finite element model of the l4-l5-s1 human spine segment including the heterogeneity and anisotropy of the discs. *Acta of Bioengineering and Biomechanics*, 17:15–24, 08 2015.
- [28] Y. Petit et al. Patient-specific mechanical properties of a flexible multi-body model of the scoliotic spine. *Medical & Biological Engineering & Computing*, 42:55–60, 2004.
- [29] M. Lund. General description of anybody. <https://github.com/AnyBody/support/wiki/General-description-of-AnyBody>. (visited on 06/06/19).
- [30] M. Christophy et al. A musculoskeletal model for the lumbar spine. *Biomechanics and modeling in mechanobiology*, 11:19–34, 02 2011.
- [31] P. Fisette. Modeling multibody systems with robotran, January 2017.
- [32] CEREM iMMC. Robotran website. <http://www.robotran.be>.
- [33] Deneire Denis and Legros Pierre-Yves. Identification de la cinématique de la colonne vertébrale sur la base des centrales inertielle imu. Master’s thesis, Ecole polytechnique de Louvain, Université catholique de Louvain, 2018.
- [34] O. Alexandrov. Gradient descent. https://commons.wikimedia.org/wiki/File:Gradient_descent.png. (visited on 04/06/19).

- [35] D. Eberly. Euler angle formulas. <https://www.geometrictools.com/Documentation/EulerAngles.pdf>. (visited on 12/05/19).
- [36] S. Godha. Performance evaluation of low cost mems-based imu integrated with gps for land vehicle navigation application. Master's thesis, University of Calgary, 2006.
- [37] D. Soares dos Santos et al. Autonomous navigation of a small boat using imu/gps/digital compass integration. *IEEE International Systems Conference (SysCon)*, 2013.
- [38] P. Tomé et al. Integrating multiple gps receivers with a low cost imu for aircraft attitude determination. *National Technical Meeting Proceedings: "Vision 2010: Present and Future"*, 1999.
- [39] S. Clerc et al. Development of the european imu for space applications. 2009.
- [40] E. Rocon et al. Application of inertial sensors in rehabilitation robotics. *IEEE 10th International Conference on Rehabilitation Robotics*, 2007.
- [41] Z. Lin et al. Development of an ultra-miniaturized inertial measurement unit wb-3 for human body motion tracking. *IEEE/SICE International Symposium on System Integration*, 2010.
- [42] A. Baghdadi et al. Hip and trunk kinematics estimation in gait through kalman filter using imu data at the ankle. *IEEE Sensors Journal*, 18(10):4253–4260, 2018.
- [43] T. Seel et al. Imu-based joint angle measurement for gait analysis. *Sensors*, 14:6891–6909, 2014.
- [44] E. Digo et al. *Tilt-Twist Method Using Inertial Sensors to Assess Spinal Posture During Gait*, volume 980 of *Advances in Service and Industrial Robotics. RAAD 2019*, pages 384–392. 01 2020.
- [45] Renata. Men silhouette. <https://fr.freepik.com/photos-vecteurs-libre/silhouette-costume>. (visited on 10/05/19).
- [46] x-io Technologies. x-imu: our original versatile imu board. <http://x-io.co.uk/x-imu/>. (visited on 03/02/19).
- [47] X. Meng et al. *Incorporating Six Degree-of-Freedom Intervertebral Joint Stiffness in a Lumbar Spine Musculoskeletal Model - Method and Performance in Flexed Postures*. *Journal of Biomechanical Engineering*, 2015.
- [48] P. De Leva. Adjustments to zatsiorsky-seluyanov's segment inertia parameters. *Journal of Biomechanics*, pages 1223–1230, 1996.
- [49] Edoarado. Anatomical planes. https://commons.wikimedia.org/wiki/File:Anatomical_Planes-en.svg. (visited on 26/05/19).
- [50] Dictionary.com. Radius of gyration. <https://www.dictionary.com/browse/radius-of-gyration>. (visited on 26/05/19).
- [51] Blausen.com staff. Medical gallery of blausen medical 2014 : Anatomical terms of location. *Wiki-Journal of Medicine*, 2014.
- [52] S. Genouel. Géométrie des masses. <http://stephane.genouel.free.fr/FT/3%20MP/12%20Cours%20-%20G%E9om%E9trie%20des%20masses.pdf>. (visited on 26/05/19).
- [53] AmesWeb. Mass moment of inertia of a cylinder. <https://www.amesweb.info/SectionalPropertiesTabs/Mass-Moment-of-Inertia-Cylinder.aspx>. (visited on 26/05/19).
- [54] Greet Kerckhofs. Lgbio2040-biomechanics: course.

- [55] A. Gomes et al. Series elastic actuator: Design, analysis and comparison. <https://www.intechopen.com/books/recent-advances-in-robotic-systems/series-elastic-actuator-design-analysis-and-comparison>. (visited on 09/06/19).
- [56] E. K. Antonsson et al. The frequency content of gait. *Journal of Biomechanics*, 18:39–47, 1985.
- [57] R. Krupenevich. Effect of low-pass filter cutoff frequencies on joint moments in walking. <https://www.forceandmotion.org/posters/effect-low-pass-filter-cutoff-frequencies-joint-moments-walking>. (visited on 29/05/19).
- [58] C. Kirtle et al. Measurement of gait kinematics. *Clinical gait analysis*, pages 39–51, 2006.
- [59] R. L. Burden et al. Natural cubic spline. In *Numerical analysis*, pages 149–150. Cengage learning, 9 edition, 2010.
- [60] Shreevardhan. Natural cubic spline interpolation in c. <https://gist.github.com/svdamani/1015c5c4b673c3297309?fbclid=IwAR2siaJUygmPph8RNn-A2oHK-9xw7ZakXxS0TenZ93MfPjflgPHxE0v3d0o>. (visited on 10/02/19).
- [61] MathWorks. Matlab. <https://nl.mathworks.com/>. (visited on 03/02/19).
- [62] Wikipedia. All models are wrong. https://en.wikipedia.org/wiki/All_models_are_wrong. (visited on 05/06/19).

UNIVERSITÉ CATHOLIQUE DE LOUVAIN
École polytechnique de Louvain

Rue Archimède, 1 bte L6.11.01, 1348 Louvain-la-Neuve, Belgique | www.uclouvain.be/epl