UCLouvain

École polytechnique de Louvain

Tumor tracking using Kalman filter on Fast MRI images

Authors: **Mélanie Ghislain, Tiffanie Godelaine**
Supervisor: **Benoît Macq**
Readers: **Antoine Aspeel, Damien Dasnoy-Sumell, Christophe De Vleeschouwer**
Academic year 2021–2022
Master [120] in Mathematical Engineering and Master [120] in Biomedical Engineering

Acknowledgements

The realisation of this project would not have been possible without the help of many people. We want to thank all of them who were directly or indirectly involved.

First, we would like to thank our supervisor, Professor Benoît Macq, who allowed us to work on this project and trusted us all along by giving us lots of freedom.

Moreover, we want to express our gratitude to Damien Dasnoy-Sumell and Antoine Aspeel for their implication during the entire realisation of this projet and their advice. They were always ready to help us and answer our questions. Moreover, we would like to thank them for their encouragement and careful review of this work.

We would also like to thank Christophe De Vleeschouwer for accepting to take from his time to read this work.

We also want to thank Lauriane Werry and Pierre Godelaine for their careful rereading and to have spent a few hours thinking about it.

Finally, we would like to extend our heartfelt thanks to our families for their constant support and encouragement throughout these five years and this final work. We are also thankful to our friends for their presence and guidance.

We would like to highlight the fact that we were very pleased to work together to achieve this project. Each of us could bring our own knowledge and qualities to construct and reach this final teamwork.

Abstract

Mobile tumors represent a major challenge for the treatment of liver and lung cancers. Radiotherapy and protontherapy are methods to treat these types of cancer. However, for the treatment to be effective, the position of the tumors needs to be known, which is difficult due to the motion induced by the respiration. To take it into account, the most used solution is the ITV-based PTV. In this context, we test to track the tumor on MRI images by first approximating it with an ellipse characterized by five parameters. These parameters are then predicted with a sinusoidal model and a Kalman filter. Segmentation algorithms (Region growing, Canny edge and Snake) are then applied when the MRI image is available to know precisely the contour of the tumor. That allows to determine whether the treatment has been correctly delivered and whether it has to be adapted due to a bad prediction. The precision of prediction and segmentation algorihtms are evaluated with the Dice coefficient. The gain is also determined by calculating the proportion of tumor and the surface of healthy cells irradiated. The principal results indicate that the Kalman filter has the best performances, i.e. the highest Dice coefficient (above 0.7). The best segmentation algorithms are the Canny edge (mean Dice from 0.7 to 0.81) and Snake (with mean Dice from 0.75 to 0.83) when the segmentation algorithm is based on a prediction returned by the Kalman filter. The main conclusion is that there is a gain in terms of surface of healthy cells irradiated when predicting the tumor (and by dilating the ellipse found) compared to the use of an ITV-based PTV treatment. The minimum gain found is around 200 mm^2 of healthy cells which are not irradiated.

Contents

Glossary

ANN Artificial Neural Network. 6

CT Computed Tomography. 1

CTV Clinical Target Volume. 4

CVA Canonical Variate Analysis. 24

DMLC Dynamic Multi-Leaf Collimator. 5

EV Explained Variance. 47

FM Fiducial Marker. 5

IC Information Criterion. 27

ITV Internal Target Volume. 4

MIMO Multiple inputs multiple outputs. 25

MOESP MIMO Output-Error State Space. 24

MRI Magnetic Resonance Imaging. 1

N4SID Numerical algorithm for subspace state space system identification. 24

PARSIM PARsimonious Subspace Identification Method. 26

PCA Principal Component Analysis. 6

PET Positron Emission Tomography. 48

PRE Percent Relative Error. 47

PTV Planning Target Volume. 4

RMSE Root Mean Square Error. 9

ROI Region Of Interest. 6

SI Signal Intensity. 14

SIM Subspace Identification Method. 24

SMAPE Symmetric Mean Absolute Percentage Error. 47

SNR Signal to Noise Ratio. 13

SVD Singular Value Decomposition. 25

TE Echo Time. 41

TM Template Matching. 6

TR Repetition Time. 41

List of Figures

List of Tables

Chapter 1

Context and objectives

1 Medical context

1.1 Cancers and radiation therapy

Cancer is a disease characterized by an abnormal division of cells that forms a non-functional mass called a tumor. This mass destroys tissues and can thus lead to the malfunction of the affected organ. At the beginning, the cancer is localized to one organ but with time, the primary cancer can spread in the body and create metastases. Depending on the progress of the cancer, two categories of treatment exist. First, systemic treatments, such as chemotherapy, act on the whole body, which allows to treat metastases. Second, loco-regional treatments treat localized primary cancer. This category includes radiation therapies such as radio- and protontherapies. Those methods send ionizing radiations on the tumor which is the treatment method we are interested in for this work. The interaction between the ionizing radiations and the tissue leads to biological effects such as damages to DNA. The treatment action also uses the difference of radiosensitivity between normal and cancer cells. Indeed, compared to healthy cells, tumor ones have difficulties to repair the damaged DNA and can no longer divide. The dose given must thus be high enough to have a high probability of controlling the tumor (in green in Figure 1.1) and a low probability of damaging healthy tissue (in red in Figure 1.1). That defines the therapeutic window [26].

Radiation treatments are composed of two major steps: the treatment preparation and the treatment delivery. During the first step, the treatment is simulated and planned. That requires anatomical information about the patient and the tumor that is acquired through computed tomography (CT) or magnetic resonance imaging (MRI). In fact, MRI is increasingly being used in radiation therapy for treatment as it provides excellent soft tissue contrast. In addition, MRI images have another advantage for the patient as they do not give an additional dose during a whole treatment as with fluoroscopy images which is an ionizing imaging technique.

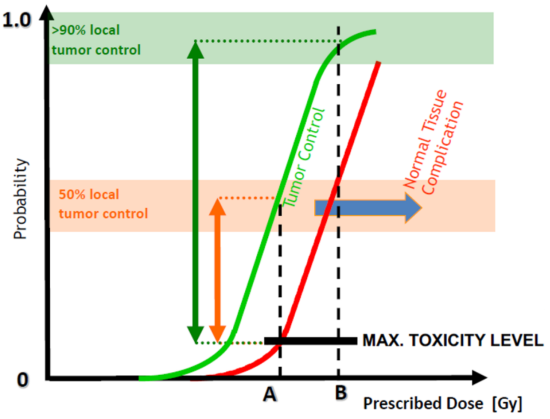

Figure 1.1: Curve of tumor control and normal tissue complication probabilities as a function of the absorbed dose. [26]

For the treatment to be effective, each cell of the tumor must be irradiated while sparing the healthy cells. Therefore, it is important to have precise information about the anatomy which is not an easy task. Indeed, the anatomy, such as the tumor position and shape, varies with time. These uncertainties can increase the risk of missing or underdosing the tumors while increasing the dose received by healthy tissues [26].

1.2 Mobile tumors motion

The location of the tumor is thus something important to know during the whole duration of the treatment to ensure the correct dose delivery to the tumor. Defining it can become more complicated when dealing with a mobile tumor. Indeed, some tumors are located close to a mobile organ and are thus subject to periodic or irregular movements. These displacements are not negligible and must be seriously taken into account during treatment to avoid missing the target and hitting too much healthy tissues. This work mainly focuses on tumors that are influenced by the respiratory motion. It is the case for lung or liver tumors.

In these cases, the tumor may move with the breathing of the treated patient. It is important to take into account the resulting movement as the aim is to irradiate the tumor while sparing the healthy tissues. Although the respiratory motion is considered as a periodic movement, irregularities in the breathing pattern can still appear during the treatment due to stress for example. As a result, the motion of the tumor will generally not be perfectly periodic. Furthermore, as it moves horizontally and vertically, the tumor can be deformed. Localising perfectly the tumor in space is thus not an easy task.

To take into account the movement of the tumor caused by breathing, some solutions were already proposed and used. This is what will be discussed in Section 3.2 and 3.3.

2 Aim of the masterthesis

The aim of the masterthesis is to track the tumor in real-time using a particle filter, more precisely a Kalman filter, on fast MRI images. Indeed, the objective is to use methods where it is possible to rapidly find parameters that fit for one specific patient. The user must be able to adjust it easily on several images. For this purpose, three steps are considered (cf. Figure 1.2 and 1.3).

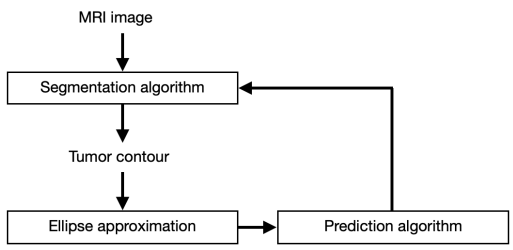

Figure 1.2: Diagram showing the three steps to reach the objective of the masterthesis.

Firstly, a segmentation algorithm can be performed on the MRI images to find the contour of the tumor. This contour is approximated by an ellipse (since we restrict ourselves to 2 dimensions) and a prediction algorithm is then applied. Once the ellipse position is predicted, we have an idea of the localisation of the tumor in the MRI image. When the MRI image corresponding to the predicted time is available, a segmentation algorithm can be applied to get the precise contour of the tumor at this time step. This precise contour will enable us to have new and accurate

information over the tumor that can be used in the following step. In fact, we can reconstruct an ellipse based on this new contour. The parameters of this new ellipse can then be used to begin the next prediction step.

The three steps are repeated at each time step. The aim is to obtain an accurate approximation at each time of the whole tumor. First, an approximation by an ellipse at each time is done thanks to the prediction algorithm, and then a refinement of this one is made thanks to a segmentation algorithm.

This scheme will be treated in 2 parts in this masterthesis, the prediction in a first time and the segmentation algorithms in a second time.

As far as the prediction is concerned, three strategies are analysed: a baseline model, a simple sinusoidal one and the Kalman prediction algorithm. In terms of segmentation algorithm, different ones will be studied and compared: the Region growing, the Canny edge, and the Snake.

One important goal of this work is to predict an ellipse and then find a precise tumor contour fast enough to be used in real-time. Indeed, the algorithm has to be usable in a few seconds. Time is important here since the prediction and the refinement are done during the treatment.

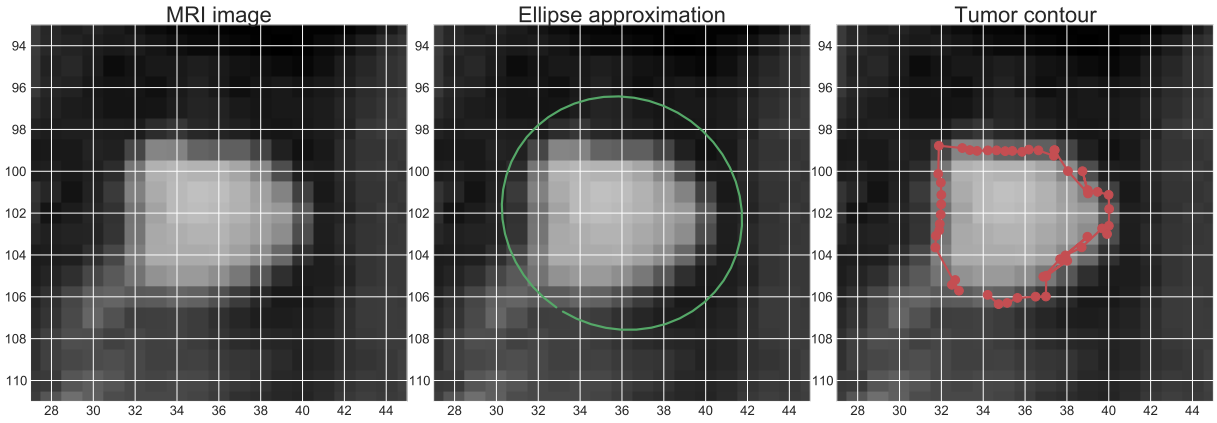

Figure 1.3: Images showing the ellipse approximation of the tumor and the tumor contour found by a segmentation algorithm.

3 State of the art

3.1 Solutions to take into account motion

First of all, diverse techniques exist to consider motion.

Among these techniques, the traditional way to take the movement of a tumor into account is to add a **safety margin** to the clinical target volume (CTV), which corresponds to the visible tumor volume where possible unvisible tumor cells are added, to get the internal target volume (ITV). Then, the ITV is enlarged to account for uncertainties in the setup to obtain the planning target volume (PTV) (as shown in Figure 1.4). Indeed, the surface to irradiate is enlarged and the safety margin is thus increased. The drawback of this method is that more healthy tissue is irradiated. That can lead to undesirable side effects such as the death of healthy cells or secundary cancers. This is why the margins must be chosen carefully as they must be large enough to take into account the motion of the tumor but not too large to avoid additional dose to healthy tissues.

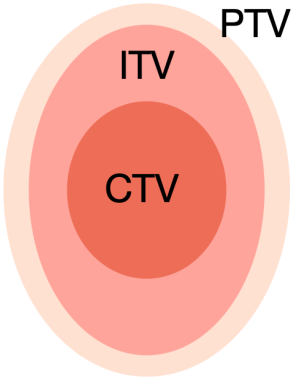

Figure 1.4: Representation of the CTV, ITV and the PTV.

3.2 Solutions to attenuate mobile tumors motion

There are also solutions to reduce the movement of the tumor due to breathing. These techniques are used during treatment.

A technique is to **regularize the pattern of breathing** by selecting some patterns that the patients have to follow. That helps the patient to have a regular and periodic respiratory movement.
In the same idea, there is the **active breathing control**. It consists of maintaining the air volume in the lungs between a certain threshold to limit the motion induced by the respiration [8]. There is also the Mechanically-assisted non-invasive ventilation that allows to modulate the breathing pattern without the use of sedation [49].

Gating is a technique currently in use that utilizes a software to observe breathing movements to define a shooting zone. This zone corresponds to the only window in which the tumor is treated.
On the contrary, once the tumor is outside this predefined zone, the radiation beam is switched off. The radiation beam is thus activated only at specific points in the respiratory cycle. One advantage of this technique is that it requires no effort from the patient [24].

A **spirometer** can also be used to measure the volume of the air contained in the lungs of the patient. Based on the volume, it is possible to know the phase in the respiratory cycle [8].

This list is not exhaustive, other methods exist. However, in this thesis we are particularly interested in methods allowing to track these tumor movements in real-time in

order to decrease the safety margin. Therefore, several recent real-time tumor tracking methods will be discussed in the next subsection.

3.3 Solutions to track mobile tumors motion

There exist several techniques to track the movements of the tumor instead of mitigating them. These techniques could, in most cases, have a beneficial role on the size of the safety margin but most of them are not yet used in practice in clinics because of their lack of reliability. To treat a tumor, an imaging technique is needed to localise it. Depending on the imaging technique used, different methods exist. Indeed, there are different kinds of images such as X-ray, ultrasound or MRI images. Among X-ray imaging techniques, two different types coexist: monoscopic and stereoscopic. The latter uses two lenses for each angle as opposed to one for monoscopic. When two lenses are used, they are placed at slightly different angles to create a depth effect in the image [38]. Most of the time, radiotherapy is combined with CT scan, an X-ray monoscopic imaging technique, but it begins to be associated with MRI. This can be seen thanks to two prototypes already present on the market, ViewRay MRIdian and the Elekta Unity system, which are called MR-guided treatment systems [8].

Another technique is called **infrared-based monitoring**. It consists of following the position of an infrared reflector using stereoscopic in-room cameras. The infrared reflector is a surrogate for the tumor position [8]. **Gold markers** implanted directly into the tumor can also be used instead of an infrared reflector and can be imaged using fluoroscopy [46]. This study shows that their position can be aquired every 0.033 seconds.

However, because MRI images are used for this work, more interest are given to what already exists in terms of tracking in MRI images. Moreover, this kind of images present lots of advantages since it has good soft-tissue contrast, does not require additional imaging dose or Fiducial Marker (FM) implantation.

The **dynamic multi-leaf collimator (DMLC) tracking system** is a method that has also been proven to track the deformation of certain tumors on 4DCT, cine-MRI and kV images. The article [22] shows two simulations: a 10 mm deformation of the primary tumor only and a whole structure containing the tumor and the lymph nodes or metastases. The goal here is to adapt directly the beam aperture to the new tumor shape based on inputs that are information about the real-time deformation. Deformable phantoms were developed to guide the aperture deformation. The method is based on an image registration software (of accuracy 1-2 mm) that automatically adapts when a new image is acquired and directly sends the computed deformation to the tracking software DMLC. Due to the time taken for the registration algorithm, the next time step is treated only when the current image registration is done. The metric uses the areas that are underexposed and overexposed to judge the accuracy of the tumor coverage. The results state that this technique reduces the under- and over-exposed regions of about 56% and 86% for single tumor and tumor system deformation respectively [22].

A recent machine, called **Linac-MR (Linear accelerator with an MRI)**, allows tumor tracking with excellent soft tissue contrast using a non-rigid registration algorithm. This method is based on finding the point of correspondance between certain MRI image

sequences despite the non-rigid deformations that the images undergo due to the breathing movement. Following this method, images are acquired before the treatment and used to select the best matches between them by minimising dissimilarities, all based on the match point. This then allows the tracking of tumor boundary during the treatment. The drawback of this method is that the algorithm requires many intensive calculations. It is thus required to optimize the algorithm in order to reduce the calculation time so that it could be used in clinics. A solution has been set up to carry out parallel computing. That allows to observe a speed up of the algorithm and to use it in real-time applications [48].

For the next two methods, images were recorded and manually processed to extract the tumor center of mass (i.e. the target position) and the region of interest (ROI) which allows us to obtain a reduced image that contains the studied structure. The principles are then different for the two methods.

The first one is based on a basic **artificial neural network (ANN)** with 3 layers (input, hidden and output layers). The input layer corresponds to the images at each moment. The output layer consists of two neurons for the x- and y-position of the target tracking. A ROI (Region of interest) around the target is used as explained above so that we do not have to focus on the whole image. The problem is dimensionally reduced thanks to Principal Component Analysis (PCA) and the learning is made thanks to a regression model and some specific activation functions. The results for this technique are good for regular breathing but less so for more irregular patterns [11].

The **template matching algorithm (TM algorithm)** is based on a template and an image comparison metric. The metric used is the cross-correlation. The aim is to find the best location of this template in the image by looking at different "viable" locations. A starting ROI size is 10x10 pixels^2. The position of the target at time 0 is identified manually as explained above. For consecutive times and frames, the template is moved in a search window of 25 pixels in each direction from the optimal position in the previous frame. In order to compute the optimal offset, the cross-correlation is computed between the image and the template. The point in the image with the highest correlation value defines the optimal offset. This process is repeated in consecutive slices to determine the offsets and therefore the positions at each time step. A multigrid search can be added to the algorithm to fill in some drawbacks. This allows a better resolution and a search in a smaller window around the maximum correlation point. Taking into account that the diaphragm has a very good correlation with the movement of the target, a correlation value of more than 0.7 is often expected. When this value is not exceeded, an estimate from a linear regression can be computed. The performance found with this algorithm is very good in all types of breathing, which leads to low tracking errors and thus to a very good real-time tracking algorithm [11].

A method based on **image correlations** was investigated for **3D motion tracking** in real-time 2D MRI series. The idea is to use a 3D image by resampling it to generate a series of 2D slices with the same orientation, slice thickness and pixel size as in the real-time 2D MRI image series. The goal is then to associate each real-time image with split 3D images with the highest normalized cross-correlation. This selection then allows to quantify the position of the tracked structure. This method was first tested on the position of blood vessels and could be used in cases where the tumor is not very visible on the MRI image as we could look at the position of adjacent structures. [10]

A new approach tracks the motion on 2D fast MRI images using manually placed trackers and transfers the tracked motion in real-time on a sequence of 3DCT images. Before the treatment, a patient specific motion model is computed using a 4DCT. Then, each tracker placed on a tissue border (the diaphragm for example) gives a breathing phase relative to the motion model. These information are then used to combine multiple deformation fields to match the breathing state at each local point, and interpolate the deformation in between. If used on a partial image, this can be fast enough to be applied in real-time during treatment. [14], [12].

Once again, this list of methods is not exhaustive. A lot of other methods exist.

3.4 Use of Kalman filter

Kalman filter is a robust algorithm that uses the measurement and inaccuracies to estimate the different states of the system. This algorithm takes benefit of the data given, knowing that there is noise and thus inaccuracy, to predict unknown values with more precision. This can be interesting in a lot of domains. At the beginning, this was thought for the application in spacecraft navigation such as positioning of GPS but its simplicity of application makes it easy to apply to many different fields. Indeed, this was already used in orbit calculation, target tracking but also in sensor data fusion, microeconomics, to predict stock prices, in instrumentation, demographic modelling, manufacturing, fuzzy logic and neural network training. This algorithm was also used in the medical field for the processing of digital image, tumor tracking, pattern recognition, image segmentation and edge detection [34].

A paper proposes an intermittent Kalman predictor to track tumors. The principle is to take measurements at no equally spaced intervals of time to focus on the times when the information contained in the measurement will be most useful for the prediction. It thus needs an adapted Kalman predictor which is compared to the regular Kalman predictor. The performance is improved since the root mean square position estimation error has decreased of 9.8% with the intermittent Kalman predictor [4].

Another paper studies the improvement that prediction can have on the treatment accuracy. It compares the performance with a linear prediction, with a neural network prediction and with a Kalman filter against a system without prediction. This analysis was made on 14 patients with an amplitude tumor movement of more than 4 mm. It uses an order between three and five for Kalman filter state vector to predict a 3D marker position. The conclusion states the relatively worse performance of the Kalman filter predictor compared to linear and ANN predictors. This low performance is expected to come from the small amount of data given to the algorithm [45].

The major drawback of Kalman filter is thus that it needs a certain knowledge about the system: the model that represents quite well the dynamics of the system we want to predict, and the measurements to have a precise prediction.

Despite this, it is used a lot to determine the location of an object in an image sequence over time, for example, tracking cars, the head of a computer user with a web-cam or to find the corners of an object in space. The approach to track an object in an image

was explained in [20] where the first guess is to look into a region centered around the previous known position of this object. Unfortunately, poor results can be found with another technique if the object moves outside this ROI. The goal is thus to look into a region centered around a more robust prediction of the center of the object. This prediction will still take into account the measurements already made knowing that they are noisy. Indeed, there are a lot of variabilities including object position, variation in brightness, and partial or full occlusion of the object which makes the position (measurement) more noisy. In this paper, three methods of prediction are compared: "no prediction" (the prediction is the last position measured), simple prediction (considering a movement in the same direction from one time step to the other) and a **Kalman filter**. The simple prediction method does not take noise into account while the Kalman filter can handle this kind of drawbacks. The results show the average prediction error of less than one pixel for all methods and states that depending on the model chosen, Kalman filter does not always have the best performance. Indeed, the results of the Kalman predictor and their accuracies strongly depend on the model chosen. Indeed, even though the kalman filter is very robust, a good choice of model can drastically increase the performance. For example, the simple prediction method is more accurate with a constant velocity model while the Kalman filter clearly outperforms the other predictors with the constant acceleration one. Nevertheless, the Kalman prediction method takes longer to recover from initial errors and thus needs a bit more time to give good results. This will also be observed in this work.

The Kalman filter is used for linear system models whether in the equation of state or measurement. Unfortunately, most systems, most behaviours are nonlinear in real life and the filters can try to adapt to this reality. This is why different versions of this famous algorithms have been thought for nonlinear systems. One example is the Unscented Kalman filter or the **Extended version of the Kalman algorithm** which can be used to predict respiratory motion in order to localise thoracic and abdominal tumors. Indeed, this kind of filter allows the system to have nonlinear state equations that represent the dynamics of the movement. The technique is here to linearize the model around an estimate of the current mean and covariance. The model on which Kalman is based here is the **local circular motion model (LCM)** that represents the natural evolution of the breathing pattern in a 2D space instead of one. In the paper, they use the first and second-order Kalman filters and try different predicting schemes. The result found is that the first-order extended Kalman filter was the most accurate for a sampling rate of 5 Hz and 0.5 s of prediction [30].

Other challenges have been taken up on fluoroscopic images acquired thanks to a fast-kV switching real-time fluoroscope using a dynamic thorax motion phantom. Indeed, the goal is to track a tumor even when it is occluded by another structure like bones, ribs or spine. The tracking is applied to a lung tumor on which a **combination of TM and an extended kalman filter** were used. This leads to the use of a Kalman filter to estimate the position of the tumor especially when the TM is not working in an efficient manner. In the paper, three different modelisation strategies were tried, applied and compared using different representations of the different states such as position, velocity, acceleration and spring motion. The performance greatly depends on the choice of the initialization parameters and the dataset was mirrored and duplicated as Kalman needs a certain amount of knowledge to work properly. The conclusion states that the physics representation with the spring motion gets the best performing results with an average

root mean square error (RMSE) of 0.44 mm which considerably improves the results of the TM alone since the mean RMSE was of 1.81 cm for a sampling rate of 1 Hz [37].

3.5 Segmentation algorithm

In the domain of biomedical image processing, image segmentation is an important step to localize the different structures. Being able to contour the tumor automatically and in real-time can improve real-time adaptative radiotherapy, in particular in the context of mobile tumor [18].

In the medical context, different segmentation methods are used and differ by their principle. One type of method is based on thresholds [36]. They are used when the region to segment has a distinctive quantifiable feature (e.g. image intensity or gradient magnitude). This region is then found by selecting the pixels whose value is in a certain interval. For example, the **Otsu's method** determines the threshold using the pixel histogram [36]. The threshold-based methods are divided into three categories: edge-based, region-based and hybrid methods. Among the edge-based methods, there are the **Canny edge detection** algorithm, that will be explored in this work, **Sobel edge detection** and **Laplacian edge detection** algorithms. Because the boundaries are not always clearly defined on medical images, these algorithms are generally not used alone but as a pre-processing step. Region-based methods, such as **region growing algorithm**, are used when the region of interest is homogeneous. The principle is to search pixels with similar feature values to form the region of interest. There are also the methods known as hybrid ones. That includes the **watershed algorithm** in which grayscale values are considered as reliefs and the gradient magnitude as elevation. The watershed lines are defined by the pixels with local maximum gradient magnitude. The pixels enclosed by the same watershed lines define a region. The problem with this algorithm is that it suffers from noise present in the image and tends to segment more objects than needed.

A second type of segmentation is based on clustering and these are the most popular techniques for medical image segmentation. Theses techniques are composed of two types of algorithms: supervised and unsupervised ones. Among the supervised classification techniques, we found the **k-nearest neighbour classifiers** (kNN), the **maximum likelihood** algorithms, the **supervised ANN**, the **support vector machines** (SVM), the **active shape models** (ASM) and the **active appearance** models (AAM). As example, the study [28] implemented a deep learning model with U-net architecture to segment tumors on MRI images. To train the model, they trained a generative adversarial network that allows to form a MRI image based on a CT scan because of the lack of MRI images dataset. The resulting Dice coefficient was around 0.75. The unsupervised techniques include the **C-mean algorithm** (CM), **the fuzzy C-means** (FCM), **iterative self-organising data analysis technique algorithms** (ISODATA) and **unsupervised neural networks**. FCM is the most used for medical applications. There are also **template matching algorithms** and **atlas-guided algorithms** that are frequently used for medical image segmentation [36]. It is the case of [18] that use a multi-template matching algorithm (MTM) to segment the primary lung tumor on fast 2D cine MRI in six locally advance lung cancer patients. The principle of this algorithm is the following: the smallest rectangle containing the tumor is found in the current image. This cropped image is then compared with the images of the training set. The one with the highest cross-correlation

is found and the contour of this image defines the current contour. In this study, they did a comparison on MRI images with three other algorithms and found that MTM algorithm has a better Dice coefficient and that the contour is found quicker, it takes about 1 ms.

A last type of segmentation is based on deformable models. Those are more flexible and can thus be used when the contour to segment is more complex. They are composed of two types of models: parametric and geometric models. The principle of parametric methods is the minimization of a moving equation for the contour that can be derived through energy function or dynamic forces. The minimum is reached at the boundaries of the region to segment. The most known energy function is composed of the internal and external energies. The internal energy is defined by the geometric properties of the contour (length, area, curvature) and the external energy contains the image information [36]. An example of parametric model is the **snake**. The problem with such method is that it is very sensitive to initial conditions and the contour may stop at a local minimum. Geometric models are based on the **level set** method. This method allows to track an evolving contour. The evolving contour is represented by a level set function and the final contour by the zero level set [36]. This evolving contour deforms with a speed that depends on the contour curvature and image features. It is comparable to the internal energy. In addition, there is another image term that allows to stop the contour at the right position, i.e. the zero level set. Algorithms based on geometric deformable models are more robust against noise and initial condition. However, as for the snake, some information about the position of the target contour is needed because the initial level set must be close to it [23].

4 Masterthesis outline

In this first chapter, the medical context and the states of the art have been introduced. In the next chapter, the method and the materials used to achieve the objective of the masterthesis will be explained. After that, the results obtained will be discussed. Finally, a conclusion will be outlined.

Chapter 2

Method

As mentioned earlier, the method is composed of three steps: a 2D approximation of the tumor with an ellipse, a prediction algorithm and a segmentation algorithm. Lastly, the goal is to combine the prediction and segmentation algorithms in order to obtain an independent technique, i.e. an algorithm that expects human initialization but which does not need the help of human intervention once it is launched.
Indeed, predicting the tumor allows to take a decision about the tumor for the predicted time step. Once the image corresponding to this time step is acquired, the precise contour of the tumor is found using the segmentation algorithm in order to minimise the treatment error, i.e. to alleviate over- and under-doses.

1 Ellipse approximation

Tumors can have many variable shapes and thus have complex contours. To simplify the contour, it is approximated by an ellipse as shown in Figure 2.2(c). An ellipse is characterized by five parameters as represented in Figure 2.2(a): the center in x and y coordinates (X, Y), the large and small half-axes a and b, and the angle of rotation α. The position of the tumor is described using the image coordinates as a reference system.

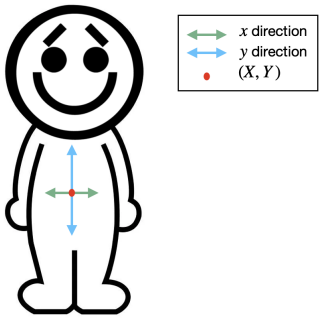

Figure 2.1: Representation of the x (in green) and y direction (in blue). The center of the tumor (X, Y) is shown in red.

The center in x and y coordinates is first determined to construct the ellipse because tumors, when tracked, are most of the time reduced to their center of mass [8]. On both the x and y axis, the mean of the minimum and the maximum value (cf. Equation (2.1)) of the contour pixel coordinates is taken. This allows to understand how the tumor moves horizontally (x direction) and vertically (y direction) over time. The direction of both movements is represented in Figure 2.1.

$$X = \frac{x_{\min} + x_{\max}}{2} \qquad Y = \frac{y_{\min} + y_{\max}}{2} \tag{2.1}$$

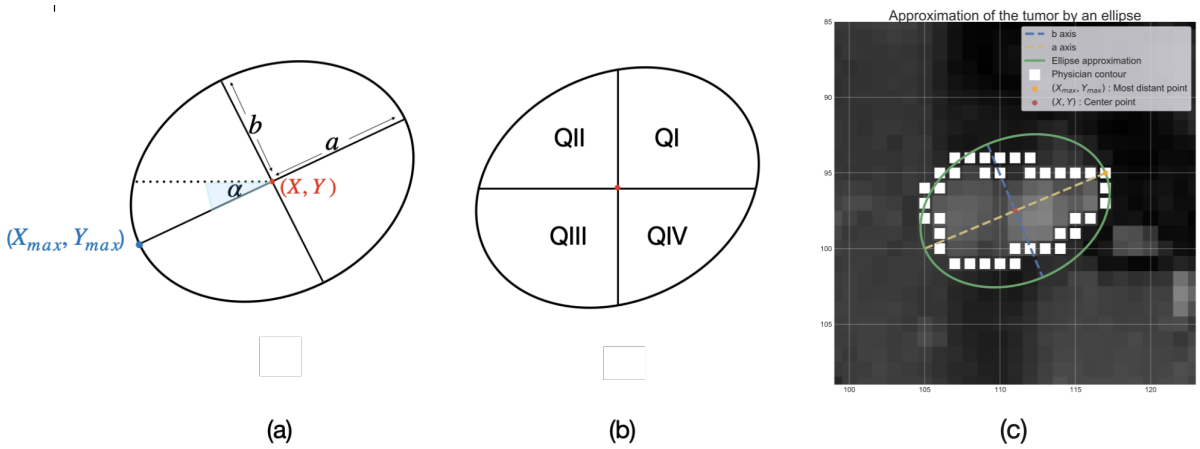

Figure 2.2: (a) Representation of the ellipse with the five parameters: the center in $x-y$ coordinates (X, Y), the large and small half-axes a and b, and the angle of rotation α. (b) Representation of the four quadrants. (c) Ellipse approximation of the tumor with the five parameters (Y, X, a, b, α) and the most distant point (X_{max}, Y_{max}) for P1S at $t = 0.4$ s.

Secondly, to determine the large half-axis a, we tried two methods namely the PCA and the most distant point method.

For the first method, a is found using the PCA. This technique is used to find a new coordinate system where variables are decorrelated and can let know the principal components that keep the most information (cf. Figure 2.3). The principal components are thus uncorrelated variables that are linear combinations of the initial ones. The first principal component is such that the variance in the projected data is maximized [33].
In our case, we can thus find the biggest axis of the ellipse by finding the axis on which the projection of the ellipse points are the most spread, i.e. the first principal component. All points are then projected on this axis in order to find the length of the big axis.

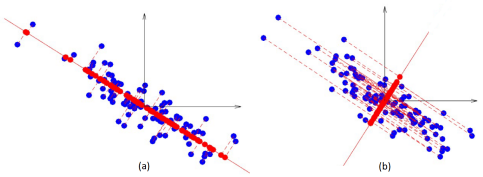

Figure 2.3: Projection of the contour points to find the axis with the maximum variance [33].

For the second method, a is build using the most distant point (X_{max}, Y_{max}) (cf. Figure 2.2(a)) of the contour from the center. To determine this point, the distance between the center and all contour points is calculated. The maximum distance corresponds to a and the corresponding contour point is (X_{max}, Y_{max}). As we suppose that the tumors do not rotate more than 90° with respiratory motion but are only deformed, the most distant point is always situated in the same quadrant of the ellipse for all time steps. Therefore, we first determine this point on the first image ($t = 1$) and the quadrant in which it is situated. For the remaining images, the most distant point from the center (X_{max}, Y_{max}) is selected only among the points situated in the right quadrant.

To find the quadrant, the coordinates of the first distant point are used. By defining the center of the ellipse as the origin point, we can pose the four different quadrants by plotting both axes. The quadrants are represented in Figure 2.2(b).

For both methods, the small half-axis b is defined by taking the line perpendicular to a and by passing through the center. All the points of the contour are then projected on this perpendicular line and the distance between the most distant point and the center determines the length of the axis (cf. Figure 2.4 for a representation).

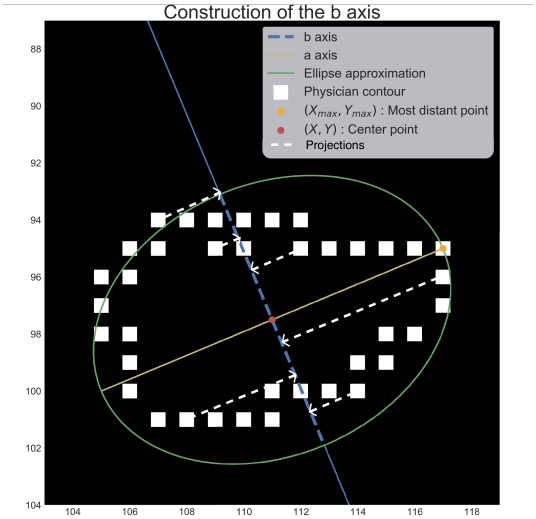

Figure 2.4: Projection of the contour points (in white) to find the length of the axis b (in blue).

The angle of rotation α is computed based on trigonometry as given by the equation (2.2):

$$\begin{cases} \alpha = \arccos(\frac{Y_{max}}{a}) & \text{if } (X_{max}, Y_{max}) \in \{\text{QI,QIII}\} \\ \alpha = -\arccos(\frac{Y_{max}}{a}) & \text{if } (X_{max}, Y_{max}) \in \{\text{QII,QIV}\} \end{cases} \tag{2.2}$$

where a is the large half-axis and Y_{max} the y-coordinate of the most distant point of the contour from the center.
The parameter α is thus defined as the angle between the horizontal and the big axis as shown in Figure 2.2(a). Depending on the position of the most distant point, the definition fo the angle of rotation is different following the quadrant considered (cf. equation 2.2) since we always define the angle positive to construct the ellipse.
As the angle of rotation depends on the big axis direction, it is thus different for both methods since the axis are defined differently. The PCA method is expected to better represent the distribution of the contour pixels in space. However, it may occur that the tumor is not entirely contained in the ellipse contour as we will see in the Chapter 5. Indeed, this method finds the axis with the highest variance and does not ensure that this axis passes through the most distant point. This can lead to the exclusion of this point from the ellipse. On the contrary, this point (X_{max}, Y_{max}) is always included in the ellipse when using the most distant point method.

An example of a tumor approximated by an ellipse is shown in Figure 2.2(c).

Noise Once the values of the ellipse parameters are found at each time step, their variation over time, namely for simplicity the trajectories $z_i(t)$, can be computed to obtain the raw data $z(t) := (z_i(t))_{i=1}^5$ where $z_i(t)$ represents one ellipse parameter at time t (index $i = 1$ and 2 for the y and x coordinate of the center, 3 and 4 for the length of axes a and b and 5 the angle of rotation). On each $z_i(t)$, a Gaussian noise of variance σ_n^2 is added to represent the noise present in the data, the noisy data $y_i(t)$ are obtained. As a result, we assume the noise on the different parameters to be independent. To determine a consistent value of σ_n, we found that the signal to noise ratio (SNR) of a FISP MRI image is around 30.6 ± 20.1 [25]. As the SNR of an image is defined as the ratio of the mean value of

the signal and the standard deviation of the noise ($SNR = \frac{SI}{\sigma_n}$ where SI is the Signal Intensity and σ_n is the standard deviation of noise), we can estimate the noise variance of the images. The noise variances for each image are summarized in Table 2.1. A mean noise variance of 0.549 are used and an analysis of the performance depending on the value of this noise will also be conducted. This value seems coherent as it is possible that the ellipse parameters vary from one or two pixels. For the angle α a smaller value is used. We assume the angle to be sure with a marge of 0.4 radians, less than for the four other parameters. Indeed, their value goes from 0.6 to 1.1 radians as the value of X position goes from -4 to 3 mm for example for one patient. This amplitude of trajectory variation is summed up in Table 5.1 and explains the reduction of noise for α for all patients.

Patient view	1C	1S	2S1	2S2	2S3	3C	3S
Noise variance	0.897	0.671	0.167	0.162	0.180	1.189	0.577

Table 2.1: Variance of the noise added on the raw data for each patient and view.

2 Prediction models

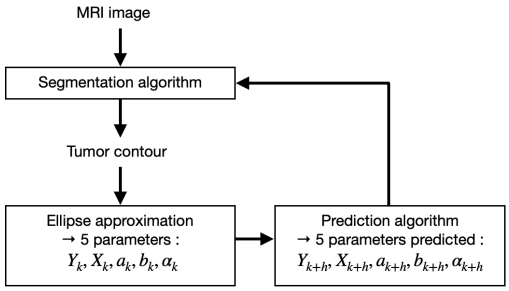

Figure 2.5: Diagram showing the three steps to reach the objective of the master thesis.

After approximating the tumor with an ellipse, the next step is to predict the position of the tumor based on the five parameters of the ellipse $(Y, X, a, b, \alpha) = (y_1, y_2, y_3, y_4, y_5)$. y_i represents the noisy measure of one of the ellipse parameter. As it is a complex problem, it is divided into several subproblems. First, the center position of the tumor in both coordinates (X, Y) is predicted. After that, the algorithm is extended to the prediction of the axes a and b as well as the angle of rotation $alpha$. The last step is to add the axis a, b and the angle of rotation α. This will give us a set of 5 variables to predict at each time step to be able to reconstruct an ellipse.

To perform the prediction, the trajectory is modelled using two kinds of technique. First, it is modelled by a simple sinusoid of the form given by the equation (2.3). Second, the prediction of the trajectory can be set with a Kalman filter. To do that, we need to model the system we are studying with a form such as equation (2.6). The unknown matrices can be found analytically (using mathematical tools, assumptions and intuition to find the general structure of the matrices) or with an identification toolbox.

2.1 Sinusoidal model

Mobile tumors, such as liver tumors, have a movement that depends on the respiratory cycle [13]. One of the ellipse parameter z_i with $i = 1, ..., 5$ at time t can firstly be approximated by a sinusoidal model of the form 2.3:,

$$\hat{z}_k = A\sin(\omega t_k + \phi) + d + b(t) = A\sin(\omega t + \phi) + d \tag{2.3}$$

where $\hat{z}_k$ is the estimation of the real trajectory at time step k, A is the amplitude, ω is the angular frequency, ϕ is the phase shift, d is the vertical shift and t_k is the time of the time step k. $b(t)$ is the noise that can be caused by measurement and is considered to be a Gaussian noise with a zero mean. Since it is an estimation, we take a mean and this last parameter $b(t)$ is put to 0. The other parameters need to be determined to construct the model.

2.1.1 Parameters determination

The frequency is firstly obtained by applying the Fourier Transform on the trajectory and taking the frequency with the highest amplitude. After that, the remaining parameters are identified.

To determine the amplitude, the period T is calculated thanks to the frequency f ($f = \frac{1}{T}$). On each period, the difference between the maximum (in green in Figure 2.6) and the minimum (in red in Figure 2.6) is computed. The half mean of these differences defines the amplitude A (in blue in Figure 2.6).

For the vertical shift, the method is quite the same except that, instead of taking the difference on each period, the mean between the maximum and the minimum is taken. The mean determines the vertical shift.
For the phase, the time at which the first maximum occurs t^* (in purple in Figure 2.6) is obtained relatively to each period and the mean time is taken. The phase is then defined as $\frac{\pi}{2} - \omega t^*$ with t^* the mean time at which the maximum occurs on a period.

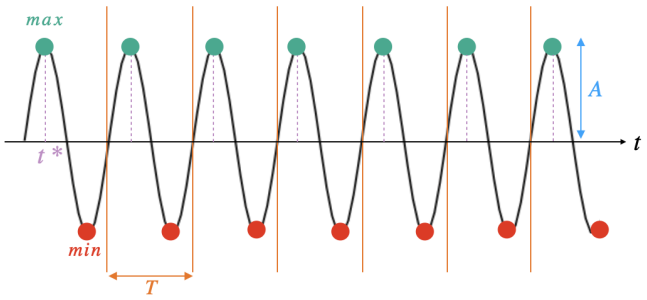

Figure 2.6: Representation of the method to determine the amplitude A.

2.1.2 Model construction

After obtaining the four parameters, the sinusoidal model can be constructed and can be used to predict the position of the tumor.
One first idea is to find these four parameters on a training set and then to apply the sinusoidal model on the rest of the data called the testing set. This gives birth to a continuous sinusoidal signal that represents the position of the tumor in time as shown in Figure 2.7. With this approach all time steps can be determined without any updates of measurement.
However, due to irregularities in the breathing pattern, a perfect sinusoidal signal is not a realistic prediction for a very long time. It is thus interesting to update these parameters over time using the data acquisition.

2.1.3 Model update

To predict the parameters, the measurements of n periods T are used. First, the period is thus identified using a certain number of samples called the training set. After that, the

four parameters are updated at each time acquisition t_k using the measurements acquired on the n previous periods as represented in Figure 2.7. These parameters A, ω, ϕ and d are used to construct the sinusoidal model that predicts the position of the tumor at step $k+h$. The predicted position being defined as

$$\hat{z}_{k+h} = A\sin(\omega t_{k+h} + \phi) + d|_{k-(n*T)}^{k} \tag{2.4}$$

Where t_{k+h} is the time of the time step $k+h$ and h is the prediction step, i.e. the time step at which we want to predict. Because the results of the prediction were not sufficient, another definition of the predicted position has been used. The article [50] states a method of adaptative sinusoidal model prediction. For that, it is possible to define the parameters of a classical model on a certain number of data (that could be a period): $z_{model}(t) = A\sin(\omega t + \phi) + d|_{t-(n*T)}^{t}$. This represents the model that will be used to predict the trajectory at future time steps. The method here evaluates this model at the predicted time but also at the actual time. This is added to the real measure at the actual time. This gives:

$$\hat{z}_{k+h} = y_k + (z_{model}(t_{k+h}) - z_{model}(t_k)) \tag{2.5}$$

Estimating the prediction error of z_{model} by testing it on data we already have enables to take into account the precision of the prediction model. Indeed, the error of the prediction at time step k $(y_k - z_{model}(t_k))$ is added on the prediction. In the article mentioned, this method seems to work properly on both regular breathing pattern and irregular ones. This will also be tested in our work to evaluate its performance.

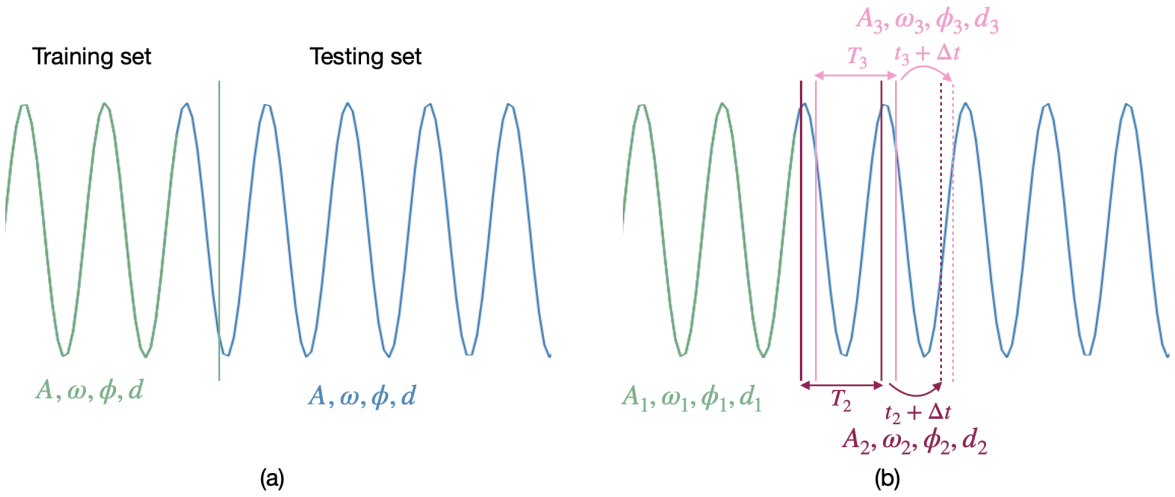

Figure 2.7: Scheme of both methods adopted. (a) Ideal sinusoidal model. (b) Model update based on one period.

2.2 Kalman filter model

2.2.1 Explanation of the algorithm

The Kalman filter was first developed in 1960 by R.E. Kalman. It allows to predict how a system is going to change by taking into account that measurements are noisy and that the system is completely known but stochastic. We can imagine a system that represents the movement of a tumor, and thus the dynamics of one, two or the five parameters representing the ellipse. The following notations are introduced: x_k is the state vector at

step k while y_k is the measurement vector at step k and z_k is the noisyless trajectories at step k. This vector in our thesis will have a size of 1, 2 or 5 to find the dynamics of the vertical movement of the tumor only, the position in $x-y$ coordinates of the tumor center and the dynamics of the ellipse movement respectively. We can thus pose:

$$y_k := \begin{cases} y_{k,1} & \text{If only the vertical movement is considered} \\ (y_{k,i})_{i=1}^{2} & \text{For the vertical and horizontal movement of the tumor center} \\ (y_{k,i})_{i=1}^{5} & \text{For the ellipse movement.} \end{cases}$$

With some assumptions of linearity and Gaussianity, the system can be defined as:

$$\begin{cases} x_{k+1} &= Ax_k + Bu_k + w_k \\ y_k &= \underbrace{Cx_k}_{z_k} + Du_k + v_k \end{cases} \tag{2.6}$$

with

$$\mathbb{E}\left[\begin{pmatrix} w_k \\ v_k \end{pmatrix}\begin{pmatrix} w_k^T & v_k^T \end{pmatrix}\right] = \begin{pmatrix} Q & S \\ S^T & R \end{pmatrix} = \begin{pmatrix} Q & 0 \\ 0^T & R \end{pmatrix} \geq 0 \tag{2.7}$$

where $x_k \in \mathbb{R}^n$, $u_k \in \mathbb{R}^{n_u}$, $w_k \in \mathbb{R}^n$, $y_k \in \mathbb{R}^{n_y}$ and $v_k \in \mathbb{R}^{n_y}$ are the system state, input, process noise, system output (measurement state) and output measurement noise respectively; and n_u, n_y and n are defined as the number of different inputs, the number of output data (in this paper it will always be comprised between one and five for the different parameters of the predicted ellipse) and the order of the system respectively. The process noise is assumed to be normally distributed with mean 0 and variance Q and the measurement noise to be normally distributed with mean 0 and variance R. So we have $w_k \sim \mathcal{N}(0, Q)$ and $v_k \sim \mathcal{N}(0, R)$. With the general system matrices being:

- $A \in \mathbb{R}^{n \times n}$ the transition matrix
- $B \in \mathbb{R}^{n \times n_u}$ the command matrix
- $C \in \mathbb{R}^{n_y \times n}$ the observation matrix
- $D \in \mathbb{R}^{n_y \times n_u}$ the direct action matrix

The equation (2.6) represents the dynamic of a general system. That explains the presence of the input, which is not the case for our system. However, we will see later that an input will be needed for the good functioning of the system identification toolbox.

Those quantities are all real given that all trajectories and measures of parameters are real.

The principle of the Kalman algorithm is to compute $\hat{x}_{k|k} := \mathbb{E}\left[x_k | Z^k\right]$ and $\hat{x}_{k+1|k} := \mathbb{E}\left[x_{k+1} | Z^k\right]$ recursively where Z^k denotes $(y_k^T, u_k^T, ..., y_0^T, u_0^T)$. To this end, the algorithm passes through the prediction-update steps iteratively as shown in Figure 2.8. It computes iteratively the estimate of the state $\hat{x}$ and the filter error covariance matrix P.

1. **The prediction step: predicted state and predicted MSE equations**

$$\hat{x}_{k+1|k} = A\hat{x}_{k|k} + Bu_k, \qquad \hat{x}_{0|-1} := \bar{x}_0 \tag{2.8}$$

$$P_{k+1|k} = AP_{k|k-1}A^T + Q - AP_{k|k-1}C^T(CP_{k|k-1}C^T + R)^{-1}CP_{k|k-1}A^T, \quad P_{0|-1} := P_0 \tag{2.9}$$

2. **The update step: update and MSE equations**

$$\hat{x}_{k+1|k+1} = \hat{x}_{k+1|k} + K_{k+1}[\overbrace{y_{k+1} - C\hat{x}_{k+1|k}}^{\text{Estimate state error}}] \tag{2.10}$$

$$P_{k+1|k+1} = P_{k+1|k} - K_{k+1}CP_{k+1|k} \tag{2.11}$$

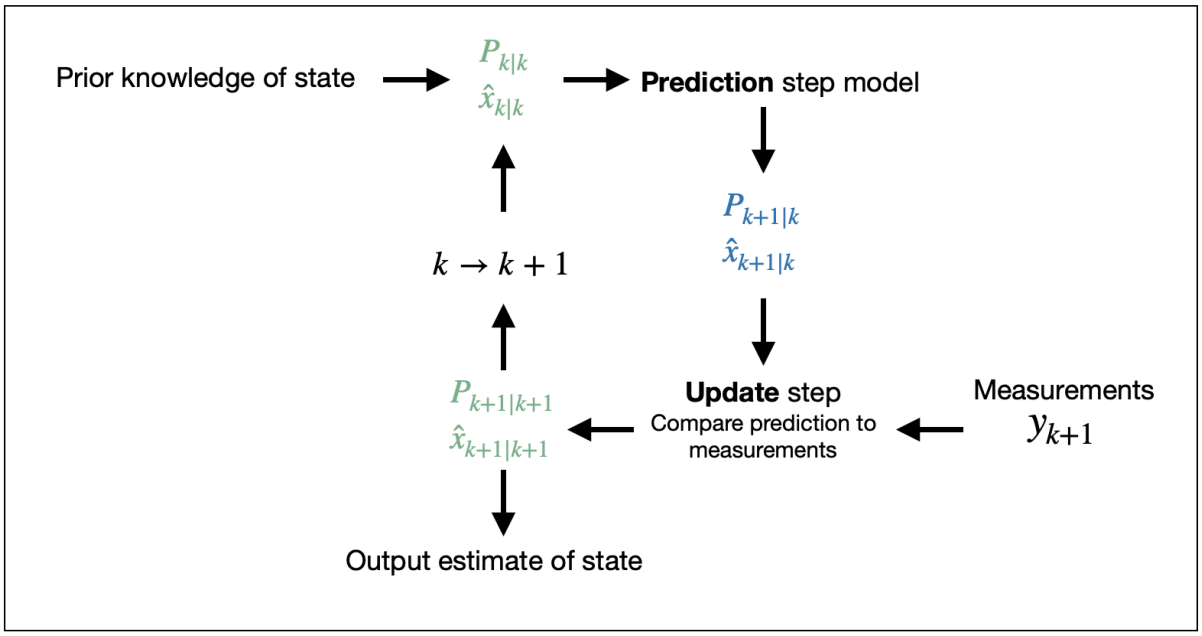

Figure 2.8: Representation of different steps of the Kalman filter.

From this we can extract the **Kalman gain equation**:

$$K_{k+1} = P_{k+1|k}C^T\left(CP_{k+1|k}C^T + R\right)^{-1} \tag{2.12}$$

We can also define C_k, the noise covariance:

$$C_k = \mathbb{E}\left[e_k e_k^T\right] = CP_{k+h|k}C^T \tag{2.13}$$

where $e_k := z_k - \hat{z}_k$.

The goal is to predict the trajectory some times in advance. In order to do that, we can use the Kalman filter theory and develop the optimal h-step ahead predictor [1]. This means that the quantity wanted is the optimal $\mathbb{E}\left[x_{k+h}|Z^k\right]$ for $h > 0$. For each state we have:

$$x_{k+h} = Ax_{k+h-1} + Bu_{k+h-1} + w_{k+h-1}$$

As we are looking for a predictor, we can develop and use the noise assumption on $\{w_k\}$:

$$\begin{aligned}\hat{x}_{k+h|k} &= \mathbb{E}\left[x_{k+h}|Z^k\right] \\ &= A\mathbb{E}\left[x_{k+h-1}|Z^k\right] + Bu_{k+h-1} \\ &= A\hat{x}_{k+h-1|k} + Bu_{k+h-1}\end{aligned}$$

By repeating this step h times to express $\hat{x}_{k+h|k}$ as a function of $\hat{x}_{k|k}$, the estimation based on observations, we have:

$$\hat{x}_{k+h|k} = A^h\hat{x}_{k|k} + \sum_{l=0}^{h-1} A^{h-1-l}Bu_{k+l} \tag{2.14}$$

To obtain the prediction of the measurement at h time steps ($\hat{z}_{k+h|k}$) instead of the prediction of the state ($\hat{x}_{k+h|k}$), we can do the following operation:

$$\hat{z}_{k+h|k} = \mathbb{E}\left[z_{k+h}|Z_k\right] = C\mathbb{E}\left[x_{k+h}|Z_k\right] = C\hat{x}_{k+h|k} = C\left(A^h\hat{x}_{k|k} + \sum_{l=0}^{h-1} A^{h-1-l}Bu_{k+l}\right) \tag{2.15}$$

We can obtain the covariance matrix of those estimation prediction at h time steps by having the same reasoning:

$$\begin{aligned}P_{k+h|k} &= \mathbb{E}\left[(x_{k+h} - \hat{x}_{k+h|k})(x_{k+h} - \hat{x}_{k+h|k})^T\right] \\ &= \mathbb{E}\left[\left(A(x_{k+h-1} - \hat{x}_{k+h-1|k}) + w_{k+h-1}\right)\left(A(x_{k+h-1} - \hat{x}_{k+h-1|k}) + w_{k+h-1}\right)^T\right] \\ &= AP_{k+h-1|k}A^T + Q \\ &= A^hP_{k|k}(A^h)^T + \sum_{l=0}^{h-1} A^lQ(A^l)^T\end{aligned} \tag{2.16}$$

2.2.2 Explanation of the model

To be able to use the Kalman filter to predict a future trajectory, a model is needed. The model must represent the trajectory of the different parameters of the ellipse. This model can be obtained by two means; analytically or based on input-output measures. This last technique will be prefered when the system becomes too complex and when there is too much unknown parameters in the model. The different steps are represented in Figure 2.9. We start from the raw data (cf. Chapter 3) that represents the real trajectory of the tumor. On this data, we add a gaussian noise to represent the uncertainties related to noisy measurement from noise image and ellipse bad approximation. The noise added has a Gaussian distribution to respect the hypotheses of the Kalman filter. After that, we divide this noisy data into three sets: training, evaluation and testing sets. The training set allows to train the model. It is composed of 60% of the data set. On this part of the dataset, we consider that the real noisyless trajectory is known. The validation set is used to choose the hyperparameters, i.e. the system order that have the best results (the smallest error of prediction). It is composed of the 20 following percent of the data. Finally, the testing set is reserved to judge the real performance of the specific parameters chosen on new data. It is composed of the last percents of the data. On the two last sets, we can compare the predicted trajectory with the real data and evaluate the performance

of our prediction methods. This also allows to identify the presence of overfitting. We apply this technique on all datasets and since one dataset corresponds to one patient, the approach is considered to be patient specific.

Two functions have been implemented. The code was based on one found in [19] that was adapted for our work. Indeed, we add the h-step ahead predictor step to the approach. The code is using two main functions, one for predicting, one for filtering. The filter function computes all important quantities such as the estimated state and covariance matrix by passing through the residual. These steps are based on the equations mentioned above. The prediction function returns the time associated to each estimation, the prediction computed thanks to equation (2.14) and the RMSE of this prediction.

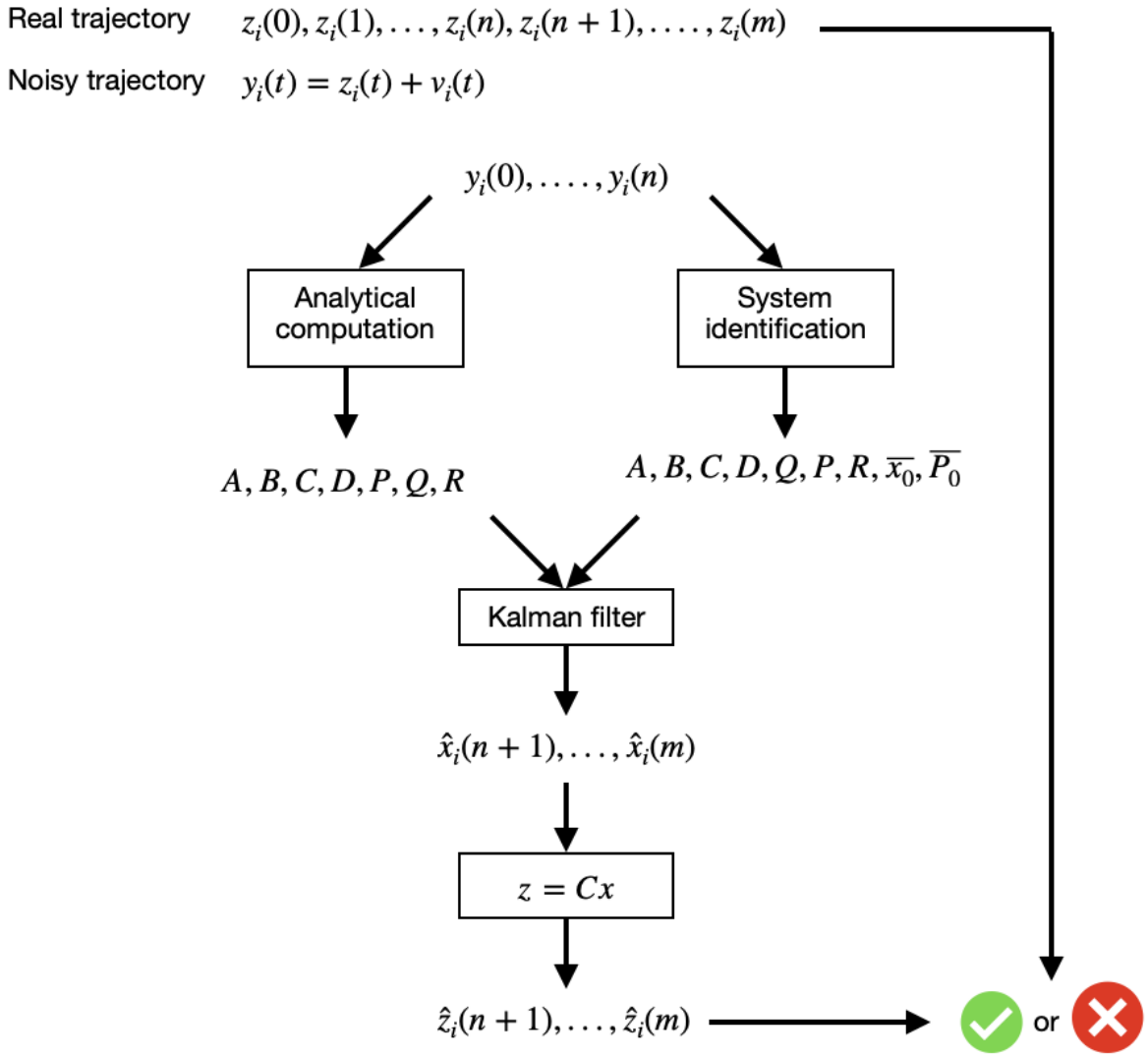

Figure 2.9: Representation of different steps of the prediction using Kalman filter.

2.3 Analytical model

2.3.1 One parameter model

One can first try to compute the model analytically for one parameter, i.e. the vertical position y_1 of the center of the tumor. Indeed, it is the parameter that is the most influenced by the respiratory motion as shown by the standard deviation of the y amplitude (c.f. Table 5.1).

The model can be represented as equation (2.6). To find the unknown parameters (the elements that compose the matrices), several assumptions are made and different conditions are used.

The first step is to determine the matrix A.

- We assume that the vertical motion induced by the respiration is periodic. To be able to obtain an oscillating trajectory, a 2D model is considered as given by the equation (2.17).
$$\begin{pmatrix} x_{1,k+1} \\ x_{1,k+2} \end{pmatrix} = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \begin{pmatrix} x_{1,k} \\ x_{1,k+1} \end{pmatrix} + w_k \tag{2.17}$$
with $w_k \in \mathbb{R}^2$, the process noise. Indeed, a linear 1D model can not give birth to a periodic solution but only to an exponential trajectory. Indeed, y is a linear combination of the states x thanks to the matrix C. A 1D continuous model can only give: $\dot{x}(t) = Ax(t) \rightarrow x(t) = Ke^{At}$ which is exponential while a 1D discrete system can give: $x_{k+1} = Ax_k \rightarrow x_k = A^k x_0$ which gives an exponential growth or only decay for the solution.

- To obtain a periodic movement, the singular values $\lambda_{1,2} = a \pm bi$ of the matrix A need to have a modulus equals to 1. This is because the singular values of a discrete model must be situated on the unit circle so that the model is periodic [47].
$$\mid \lambda_{1,2} \mid = \sqrt{a^2 + b^2} = 1$$

- In addition, based on the equation (2.17), we know that $A_{12} = 1$ and $A_{11} = 0$. Which already gives $A = \begin{pmatrix} 0 & 1 \\ A_{21} & A_{22} \end{pmatrix}$.

- Moreover, it is known that the sum of the singular values must be equal to the trace of the A matrix.
$$\lambda_1 + \lambda_2 = \text{trace}(A) = A_{11} + A_{22} = 2a$$

- The product of the singular values equals the determinant of the A matrix.
$$\lambda_1 \lambda_2 = \det A = A_{11}A_{22} - A_{21}A_{12} = (a + bi)(a - bi) = a^2 + b^2$$

By combining these conditions, we obtain the following matrix:

$$A = \begin{pmatrix} 0 & 1 \\ -1 & 2c \end{pmatrix}$$

Only one parameter c remains undefined. To find a value that models as well as possible the system, we try a value proportional to the frequency of the trajectory because we want

to model a periodic signal. The training set can be used to find the value of c.

We then assume that the system is free, i.e. without input. Indeed, the movement of the tumor does not depend on an external input as it is due to the respiration. The matrices B and D are thus null: $B = 0_{2\times1}$, $D = 0$.

After that, because we want to observe the state $x_{1,k+1}$ and to simplify the system as much as possible, we define the matrix C as

$$C = \begin{pmatrix} 1 & 0 \end{pmatrix}$$

This allows us to give a sense to our states since we link them directly to the measurements. This will ease the following reasoning.

The matrix Q, the state covariance matrix, $w_k \sim \mathcal{N}(0, Q))$, is related to the imprecision of the model and is a covariance matrix which is thus positive-semidefinite and symmetric. Moreover, since our observability matrix C is $\begin{pmatrix} 1 & 0 \end{pmatrix}$, the state $x_{1,k+1}$ represents the vertical position. We can thus deduce and construct this matrix Q based on the information contained in the measures. Since the two states represent the vertical position one step apart, they have the same information content and the diagonal of Q has the same value q. Finally, we suppose uncorrelated noises.

$$Q = \begin{pmatrix} q & 0 \\ 0 & q \end{pmatrix}$$

Based on its definition, q is determined by simulating the model and obtaining $y = C(Ax) + v$ with $x = \begin{pmatrix} x_{1,k} \\ x_{1,k+1} \end{pmatrix}$, and by taking the variance of the difference between the output of this model and the real trajectory: $q = Var(y - z)$ with z the real noisyless trajectory and y the simulated output from the model.

The matrix R, which is the input measurement noise ($v_k \sim \mathcal{N}(0, R)$), is here a one by one, so a real as the number of output is 1. In the same way as Q, it is the covariance matrix of a Gaussian noise and it is thus positive-semidefinite and symmetric. It is defined as $R = \sigma_n^2 \in \mathbb{R}^{n_y \times n_y}$ as it gives the uncertainties on the measures. This includes the noise in the image that we might try to estimate since the measures are done using a MRI scanner. The value of σ_n^2 is taken as 0.549 as explained previously in Section 1 of this Chapter.

The matrix P_0 ($x_0 \sim \mathcal{N}(x_0, P_0)$) gives information about the uncertainties of the initial state x_0. As Q and R, it is a covariance matrix and is thus positive-semidefinite and symmetric. We consider here that it corresponds to the measurement noise since our states correspond in a physical sense to the vertical position and thus the initial state is a measurement. The value is here $P_0 = R\mathbb{I}_n$.

2.3.2 Two parameters model

We then tried to find a model analytically to represent the dynamics of two parameters: the vertical y and the horizontal x positions of the tumor center.

The model is represented by the same equation as the one parameter model (cf. equation (2.6)). Again, we need to make assumptions and use conditions to find the unknowns.

To find the matrix A, the following known information is used:

- Because the horizontal motion of the tumor induced by the respiration is assumed to be periodic such as the vertical motion, a 4D model represented by the equation (2.18) is considered.

$$\begin{pmatrix} x_{1,k+1} \\ x_{1,k+2} \\ x_{2,k+1} \\ x_{2,k+2} \end{pmatrix} = \begin{pmatrix} A_{11} & A_{12} & A_{13} & A_{14} \\ A_{21} & A_{22} & A_{23} & A_{24} \\ A_{31} & A_{32} & A_{33} & A_{34} \\ A_{41} & A_{42} & A_{43} & A_{44} \end{pmatrix} \begin{pmatrix} x_{1,k} \\ x_{1,k+1} \\ x_{2,k} \\ x_{2,k+1} \end{pmatrix} + w_k \tag{2.18}$$

  With $w_k \in \mathbb{R}^4$, the process noise. The 4D model is the combination of two 2D models, each one representing one of the movement. It is interesting to mention that it would be possible to obtain two periodic movements using a 2D model represented by $\begin{pmatrix} x_{1,k+1} \\ x_{2,k+1} \end{pmatrix} = A \begin{pmatrix} x_{1,k} \\ x_{2,k} \end{pmatrix}$. Since it is a 4D system, we have 4 eigenvalues denoted: $\lambda_{1,2,3,4}$.

- In order to have a periodic motion, the singular values of the matrix A must have a unit modulus in the same way as the one parameter model. We assume that there are two double eigenvalues with modulus 1 since we want to obtain two periodic trajectories. We consider thus the eigenvalues to be noted as: $\lambda_{1,2} = a \pm bi$ and $\lambda_{3,4} = c \pm di$.

$$|\lambda_{1,2}| = \sqrt{a^2 + b^2} = 1 \qquad |\lambda_{3,4}| = \sqrt{c^2 + d^2} = 1$$

- Based on the equation of system (2.18), it is trivial that $A_{11} = A_{13} = A_{14} = A_{31} = A_{32} = A_{33} = 0$ and $A_{12} = A_{34} = (a^2 + b^2)(c^2 + d^2) = 1$

- The conditions on the eigenvalues are the same as the ones used for the one parameter model:

$$\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 = \text{trace}(A) = A_{11} + A_{22} + A_{33} + A_{44} = 2(a + c)$$

$$\lambda_1\lambda_2\lambda_3\lambda_4 = \det(A) = A_{23}A_{41} - A_{21}A_{43} = 1$$

Combining all the information, we obtain the following matrix:

$$A = \begin{pmatrix} 0 & 1 & 0 & 0 \\ A_{21} & A_{22} & A_{23} & A_{24} \\ 0 & 0 & 0 & 1 \\ A_{41} & A_{42} & A_{43} & A_{44} \end{pmatrix}$$

Even when using all the conditions, a lot of undefined parameters remain. We can either assume that both systems can be separated and are independent, which gives two systems as the one parameter (with $A_{23} = A_{24} = A_{41} = A_{42} = 0$). However, this is a strong and not realistic assumption as both movements are induced by the same respiration and are thus correlated. In this last case, the eight unknown parameters stay unknown and this becomes complicated to conclude. That is why we decide to use a system identification to find the two parameters model and the five parameters model (the matrices of the system of equation (2.6)).

2.4 System identification

The system can be represented as equation (2.6). The different matrices and vectors have each a defined role as already stated in Section 2.2 of this Chapter.

The goal of the system identification phase is to retrieve the principal matrices of the system A, B, C, D, Q, R, x_0 thanks to the study of signals. Those signals correspond to the trajectories of the ellipse parameters extracted from the MRI images on which noise is added as explained in Section 1 of this Chapter. They are given to the system identification toolbox in order to identify a state-space model directly from input-output data. In a sense, an input is not obvious in this context as we have only trajectories of output and a free system. Indeed, the movement of the tumor does not depend on an external input as it is due to the respiration. But an input is set here as it is required for a proper functioning of the identification toolbox. To allow the use of an input, we can consider it is an affine term that can be added without setting the system in an unstable state. We consider $u(t)$ as the delta function of value 0.1 at $t = 0.4$ s, like an offset term. The input signal must be quasi-stationary [5] and must be persistently exciting of order minimum $f + p$, where f and p stand respectively for future and past horizons. A delta function is persistently exciting of order infinite since its spectrum is the constant function and any continuous function can be written as an infinite sum of dirac delta. We also tested with $u(t)$ equals to a step function of amplitude 0.1 shifted to the right of one unit of time and $u(t)$ equals to a constant. We tested using these input functions but no prediction was obtained.
Once the model is obtained, a Kalman Filter is used to filter and predict the trajectory as explained in the previous section. One must know that the system is unique up to a similarity transformation ($A' = BAB^{-1}$, where A and A' are similar matrices).

For this problem, Subspace Identification Methods (SIMs) are used since the goal is to identify a state-space model. To identify transfer function models, Prediction Error Methods will be prefered. To identify the state-space models, the subspace identification algorithms are using N4SID, MOESP and CVA (which are traditional methods), or the parsimonious methods [3].

An option was developed with those models to put the direct action matrix D to zero. This can be useful for a system without a direct feedforward loop, so no direct relation input-output is dressed. A certain knowledge of the system has to be known for this. If this is not mentioned, the matrix D will be computed in the same way as the other command and observability matrices. Moreover, the stability can be imposed thanks to a parameter in the traditional method. This option is not possible for the parsimonious methods. The stability implies that the norm of the eigenvalues is smaller than one and they lie in the unit circle (they can be imaginary numbers). An extension of the observability matrix is computed to impose the stability when the returned A is unstable. The command matrix is not correct anymore since it is based on the previous A. It is thus recomputed too. This process is still dangerous since it can lead to identification errors. Imposing stability or reducing D to a null matrix is a consequent decision.

In this process, the D matrix values will be directly set to 0 and the stability will be required. The reasons are the unexpected input so we assume the measures not to be influenced by it. In addition, the unstability can lead to signals that explode in values and grow very fast. The expected signals are for two of them periodic and the axes as well as

the angle are measures that do not evolve exponentially. The stability is thus something we expect.

2.4.1 SIMs methods

The SIMs methods can be applied on open-loop systems with input data independent of past noise. These methods include N4SID, MOESP and CVA. Their general principle consists of three major steps with sometimes a pre-processing phase. These are the projection or regression, the model reduction, and the parameter estimation.

- The projection or regression is the step during which a least squares regression or projection is performed to estimate one or several (up to f, the future horizons of the system identification) high-order models.
- The model reduction is the process of reducing the order of the subspace. It gives an estimation of the state of the system.
- The parameter estimation is the step of computation of the system general matrices such as A, B, C, D, Q, R and S. By using the observability matrix of the reduced system, we come to a solution [41].

The main differences can not only be stated between the traditional and the parsimonious methods but also between the traditional methods themselves.

MOESP states for MIMO Output-Error State Space model identification approach. The main point of that method is that it uses an QR factorization. This factorization is a pre-processing step that decomposes a matrix into a product of an orthogonal and an upper triangular matrix and allows to reveal its special structure (column or row). It is sometimes followed by a Singular Value Decomposition (SVD), especially when the acquisition is done with errors. The SVD consists of the factorization of the matrix into a product of unitary matrices composed of the singular vectors and a diagonal matrix containing the singular values. The last part is to find the solution of an overdetermined set of equations. When a compatible solution can not be found, it is solved in a least squares sense and it gives an approximate solution. The factorization is made on the matrix pair made of the Hankel matrices $U_{1,i,n}$ (defined underneath) and $Y_{1,i,n}$ that are constructed from the Markov parameters. Markov parameters are the impulse reponse terms of the state-space model defined as $h(n) = CA^nB$ with $n > 0$ [27]. After the QR factorization, the SVD is made on a matrix computed thanks to the Toeplitz cross covariance matrix H_i defined as [51]:

$$U_{0|i-1} = \begin{pmatrix} u_0 & u_1 & u_2 & \cdots & u_{j-1} \\ u_1 & u_2 & u_3 & \cdots & u_j \\ \vdots & \vdots & & & \vdots \\ u_{i-1} & u_i & u_{i+1} & \cdots & u_{i+j-2} \end{pmatrix}, H_i = \begin{pmatrix} D & 0 & 0 & \cdots & 0 \\ CB & D & 0 & \cdots & 0 \\ CAB & CB & D & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ CA^{i-2}B & & & \cdots & D \end{pmatrix}$$

N4SID is noted for Numerical algorithm for subspace state space system identification. As the MOESP algorithm, it uses QR factorization and the Singular Value Decomposition, but here no Toeplitz matrix must be computed, which makes it numerically more stable. Moreover, it is stated that this kind of algorithm is always convergent. Finally, a zero or

non-zero initial state does not matter for this algorithm. The first step is to construct the block Hankel matrix and to compute the QR factorization of this matrix. A subspace is found from the projection of future outputs onto past and future inputs and past outputs. $Z_i = Y_{i|2i-1}/(U_{0|2i-1}, Y_{0|i-1})$ and $Z_{i+1} = Y_{i+1|2i-1}/(U_{0|2i-1}, Y_{0|i})$ where $A/B := AB^T(BB^T)^{-1}B$. The necessary information is retained in those new matrices and they are the basis to find the state space matrices. The SVD can be used to know the order as it will correspond to the number of non-zero singular values of the matrix T. This last matrix can be any rank deficient matrix for which the column space is the same as the one of the observability matrix Γ_i:

$$\Gamma_i = \begin{pmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{i-1} \end{pmatrix}$$

The last step is to determine the system matrices in a least squares sense thanks to the different decompositions made. A and C are determined immediately and B and D follow them. The covariance matrices are determined thanks to ρ, the residuals of the least square solution. $\frac{1}{j}(\rho\rho^T) \approx \begin{pmatrix} Q^s & S^s \\ (S^s)^T & R^s \end{pmatrix}$. The variable i corresponds to the number of data and as i grows larger, the approximation error grows smaller. For deterministic subsystem, the parameters can be guessed exactly while for stochastic system, an approximation can be computed. However, for infinite i, the stochastic subsystem can be determined exactly [39].

CVA is the acronym of Canonical Variate Anlaysis and is the third traditional method we have tested and we explained. The state corresponds to the Markov state of process. The first step of this method is to determine the canonical states of the process and by selecting the first n ones, an optimal state for a model of order n is thus found. This step allows to reduce the order state. A simple regression can then be used to obtain the state equation. The difference is made between this method and the others in the order of the processes. CVA can also be used and can perform when the order of the system is uncertain [32]. One must be aware that Markov parameters may sometimes be difficult to compute.

2.4.2 Parsimonious methods

PARSIM comes from PARsimonious Subspace Identification Method. Four types exist: PARSIM-P, PARSIM-S, PARSIM-K and PARSIM-E.

- PARSIM-P means Parallel PARSIM. This method removes the non-causal terms of the input vector u and involves least squares problems in parallel [43].
- The S of PARSIM-S states for sequential. It is based on the same principle as PARSIM-P except that least squares problems are sequential [44].
- PARSIM-E means PARSIM with Innovation Estimation which is based on the estimation of the innovation terms to take into account the past innovation in the least squares regression [40].

- PARSIM-K estimates system matrices from the predictor form and is also based on least squares problems. The difference is that the matrix K is computed along with the B_K, D, x_0 while other algorithms compute it in a separate step [40].

There are 6 methods that could thus be used for our identification and based on the information found in articles, only three were compared. Using [42], we know that the Parsimonious Methods do not return the covariance matrices. However, this is an important element in the Kalman algorithm. It could be possible to compute them analytically following certain methods like we did for the analytical way (cf. Section 2.3) but the number of parameters would start to become high for large orders. Moreover, it has been observed that the PARSIM-K method has good performance in the training phase but was less robust to noise in the validation phase [3]. Therefore, we decide to compare N4SID, MOESP and CVA methods.

Most algorithms required some a priori knowledge over the system to work properly, such as the fact that D is null or the order of the system. With those traditional methods, there is an option to find the best order of model. This option is called Information Criterion (IC) [3]. The basic idea of this criterion is to find an order not too high to alleviate overfitting but not too low so that the complexity of the model can be taken into account and so that it stays robust to errors. It consists of finding the minimum of a likelihood function on which a penality term for high orders is added. However, this approach is not the central point of this work. We prefer to follow a more intuitive approach to find the order of the system.

2.4.3 System identification toolbox

For the implementation, we used the Systems Identification Package for PYthon namely SIPPY. When using CVA, N4SID and MOESP methods, it returns the following matrices: $A, B, C, D, A_K, B_K, K, ts, G, x0, Q^s, R^s, S^s, V_n$ [42]. For a system of order n, the size of the matrices returned by the system identification is given in Table 2.2:

A	B	C	D	A_K	B_K	K	ts	x_0	Q	R	S	Vn
$\mathbb{R}^{n\times n}$	$\mathbb{R}^{n\times n_u}$	$\mathbb{R}^{n_y\times n}$	$\mathbb{R}^{n_y\times n_u}$	$\mathbb{R}^{n\times n}$	$\mathbb{R}^{n\times n_u}$	$\mathbb{R}^{n\times n_y}$	$\mathbb{R}$	$\mathbb{R}^{n_y}$	$\mathbb{R}^{n\times n}$	$\mathbb{R}^{n_y\times n_y}$	$\mathbb{R}^{n\times n_y}$	$\mathbb{R}$
n^2	$n_u n$	$n_y n$	$n_y n_u$	n^2	$n_u n$	$n n_y$	1	n_y	n^2	n_y^2	$\mathrm{n}_y n$	1

Table 2.2: Size of the matrices of the model.

The different responses can be defined as:

- A, B, C and D are the matrices of the principal system (cf. equation (2.6)) as defined above.
- A_K and B_K are the matrices of the predictor form:

$$\begin{cases} x_{k+1} &= A_K x_k + B_K u_k + K y_k \\ y_k &= C x_k + D u_k + e_k \end{cases} \tag{2.19}$$

  where the following relations hold:

$$A_K = A - KC$$
$$B_K = B - KD$$

This is obtained thanks to the innovation form:

$$\begin{cases} x_{k+1} &= A_K x_k + B_K u_k + K e_k \\ y_k &= C x_k + D u_k + e_k \end{cases} \tag{2.20}$$

and replacing e_k by $y_k - \hat{y}_k$, where $\hat{y}_k$ is the estimated noisy measurement.

- K is the steady-state Kalman filter gain obtained from Algebraic Ricati Equation equation (2.9) and equation (2.12).
- ts is the sampling time of the identified discrete system.
- G is the linear state-space model.
- x_0 is the initial state estimate.
- Q, R and S are the covariance matrices of the system, more developed: $\begin{pmatrix} Q & S \\ S^T & R \end{pmatrix} = \mathbb{E}\left[\begin{pmatrix} \rho_w \\ \rho_v \end{pmatrix}\begin{pmatrix} \rho_w^T & \rho_v^T \end{pmatrix}\right]$ where ρ_w and ρ_v are the estimated sequences of the state noise and output measurement noise (w and v).
- V_n represents the estimated error norm that is computed as:

$$V_n = \frac{1}{2L} tr(\epsilon\epsilon^T)$$

where tr is the trace operator on matrices and $\epsilon = y - y_{model}$ is the prediction error sequence where y_{model} is the output of the identified model while y is the real measures given to the system identification toolbox.

Because Q and R are symmetric as there are covariance matrices, the number of undetermined parameters is equal to $\frac{n^2+n}{2}$ instead of n^2. In addition, A_K is obtained thanks to A and C while B_K from B and D. If K is computed, A_K and B_K can be deduced. This removes $n^2 + n_u n$ parameters. D is stated to be equal to the null matrix of appropriated size. This gives $n_y n_u$ parameters that have not to be computed.
Based on Table 2.2 and the previous information, the total number of parameters n_{tot} to compute is:

$$n_{tot} = n_u n + 3n_y n + n_y n_u + 2 + \frac{3n^2 + n}{2} + \frac{n_y^2 + 3n_y}{2}$$

Considering that the input size $n_u = 1$ and output size $n_y = \{2, 5\}$ depending on the number of ellipse parameters we predict (Y, X, a, b, α), the total number of parameters for several orders are summarized in Tables 2.3 and 2.4.

- For $n_y = 2$ (system of two parameters (X, Y)): $n_{tot} = n + 6n + 2 + 2 + \frac{3n^2+n}{2} + \frac{2^2+6}{2} = \frac{3n^2+n}{2} + 9 = \frac{3n^2+15n}{2} + 9$

Order n	2	3	4	5
Number of parameters n_{tot}	30	45	63	84

Table 2.3: Number of parameters to determine for $n_y = 2$ and $n = \{2, 3, 4, 5\}$.

Order n	2	3	4	5
Number of parameters n_{tot}	66	90	117	147

Table 2.4: Number of parameters to determine for $n_y = 5$ and $n = \{2, 3, 4, 5\}$.

Patient	**1C**	**1S**	**2S1**	**2S2**	**2S3**	**3C**	**3S**
Training set	130	130	108	108	108	132	132
Test / validation set	42	42	36	36	36	44	44

Table 2.5: Size of the training, test and validation sets for each patient and view.

- For $n_y = 5$ (system of five parameters (X, Y, a, b, α)): $n_{tot} = n + 15n + 5 + 2 + \frac{3n^2+n}{2} + \frac{5^2+15}{2} = \frac{3n^2+33n}{2} + 27$:

This number of parameters that the system identification has to compute can be linked to the number of data the system identification will have at disposal. Indeed, there is a certain relation between those two quantities. For the system identification to construct a robust and precise model, it is recommended to have much more data than the number of parameters to compute. A ratio of 10 is advised. As already said, the system will be identified thanks to a set of data called the training set. This training set is composed of 60% of the data set. That corresponds to a mean of 121 images. Therefore, there is a mean of 242 and 605 samples in the training set for $n_y = 2$ and $n_y = 5$ respectively. The exact value for each patient is specified in Table 2.5.

3 Segmentation algorithms

Three different segmentation algorithms (Region growing algorithm, Canny edge detection and Snake algorithm) are tested and compared to find the real contour of the tumor. This section allows to explain the use of different segmentation algorithms for the tracking of tumors in MRI images. The goal of the refinement given by the segmentation algorithm is to be able to judge the quality of the treatment. By measuring the excess doses, the treatment could be adapted in the next time steps.

The parameters for each segmentation algorithm are chosen based on the results on the training set. The optimal parameters are then used on the testing set to compare the performance of the segmentation algorithms on data that were never used previously.

For this section, we consider for simplicity that $(\hat{z}_1, \hat{z}_2, \hat{z}_3, \hat{z}_4, \hat{z}_5) = (Y, X, a, b, \alpha)$.

3.1 Region growing algorithm

As already said, Region growing is a region-based segmentation method. The main principle is to add pixels that meet a certain condition to a surface.

First, a seed pixel is selected. The seed point is usually put in the center of the ROI. To automate the algorithm, the seed pixel is defined by the center of the predicted ellipse (X, Y) as shown in Figure 2.10. Indeed, we assume that the predicted parameters (Y, X, a, b, α) form an ellipse that is a good approximation of the tumor.

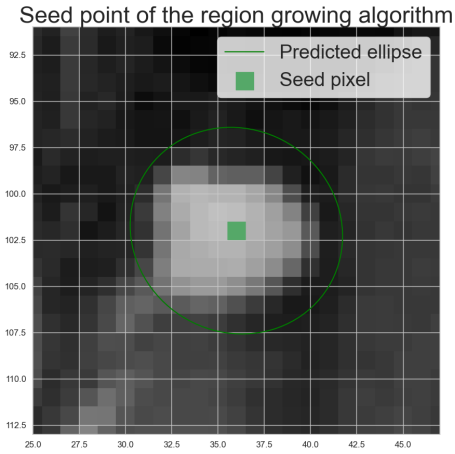

Figure 2.10: Seed point of the region growing algorithm for P1C at $t = 0.4$ s.

After selecting the first pixel of the region as explained just above, this area, called "growing region", is extended with the neighbouring pixels that meet a certain condition. As it is often the case, this condition is based on the pixel intensity. To be added to the surface, the pixel must have a gray value in a certain interval. This interval is defined by:

$$[(\text{mean gray value of the pixels} \in \text{to the surface}) - \epsilon \times (\sigma \text{ of the pixels} \in \text{to the tumor}), (\text{mean gray value of the pixels} \in \text{to the surface}) + \epsilon \times (\sigma \text{ of the pixels} \in \text{to the tumor})] \tag{2.21}$$

with σ the standard deviation, ϵ the percentage of the standard deviation taken and $\epsilon \times (\sigma$ of the pixels $\in$ to the tumor) the margin of acceptance in the region. To determine the standard deviation, the pixels that belong to the tumor are found on the first five MRI images thanks to the physician's contours and the standard deviation of their gray value is taken. Indeed, this gives an idea about how the intensity of the pixels in the tumor varies. To adapt as much as possible the condition to enter in the growing region and thus the parameter of the condition, we tried several percentages of the standard deviation found on a training set composed of the five first MRI images and their corresponding physician's contour.

Another condition can also be added, e.g. a condition based on the gradient. In our case, we decided to define a stopping criterion for the Region growing algorithm relative to the surface of the region, more precisely to the number of pixels that compose the region. When this number is higher or equal than the mean number of pixels belonging to the ellipse approximated the tumor times 1.2, an error margin, the algorithm stops. The mean number of pixels is found using the five first physician's contours that were filled. This stopping criterion was chosen to take into account the fact that the difference of gray values between the tumor and the background of the MRI image may be low. That prevents the surface from growing too much if the condition is respected for pixels that do not belong to the tumor. Indeed, without this criterion, it may occur that the surface continues to grow outside the area of interest where the tumor is located. A solution to avoid this would have been to use a ROI. In addition, the order of the neighbouring pixels to check is always the same: the upper pixel, the right one, one below and the left neighbouring pixel.

For the implementation, we first start by fixing the seed point knowing the parameters of the predicted ellipse. We then create two matrices of the same size as the image. One allows to indicate the pixels that belong to the growing region and the other the pixels that have already been checked, meaning that we know whether they are respecting the condition or not. We also create a vector that contains the neighbouring pixels to check. After that, the algorithm begins by adding the neighbouring pixels (those which are not already present in the vector of pixels to check) to the vector. Then, all the pixels contained in the list are verified and the ones that verify the condition are added to the region and its neighbours are added to the list of pixels to check. The algorithm stops once this last vector is empty or if the condition on the surface is not respected anymore.

Because the physician's contour is always a bit bigger than the exact contour of the tumor, to be sure that it contains all tumoral cells, we tried to dilate the obtained region to see whether it gives better results. The same training set, composed of the five physician's contours, can be used to determine if it is the case. The dilation is a process of combining two sets using the vector addition of set elements. This allows to change the size of an object without changing its general shape using a kernel. The kernel is thus added to the object of interest and the object is thus enlarged.

3.2 Canny edge detector

Canny edge detector is an edge-based method. This involves finding the contour by keeping some pixels of the image based on its gradient.

We firstly crop our image to only have the ROI, i.e. a region around the tumor and containing it. The center of the ROI is defined by the center of the predicted ellipse. Cropping the image allows us to restrict the domain of our segmentation algorithm and save time which is important as it is for a real-time application. Indeed, we decrease the domain of search of the algorithm instead of letting it look at the entire image. In addition, that prevents from taking the wrong contour if there are similar objects in the image or other borders or changes of intensity.

The second step is to find the gradient magnitude and direction. The gradient in both directions G_x and G_y are computed using the Sobel operator. They are then used according to 2.22, 2.23:

$$|G| = \sqrt{G_x^2 + G_y^2} \tag{2.22}$$

$$\theta(x, y) = \arctan(\frac{G_y}{G_x}) \tag{2.23}$$

The magnitude gives an image represented in Figure 2.11.

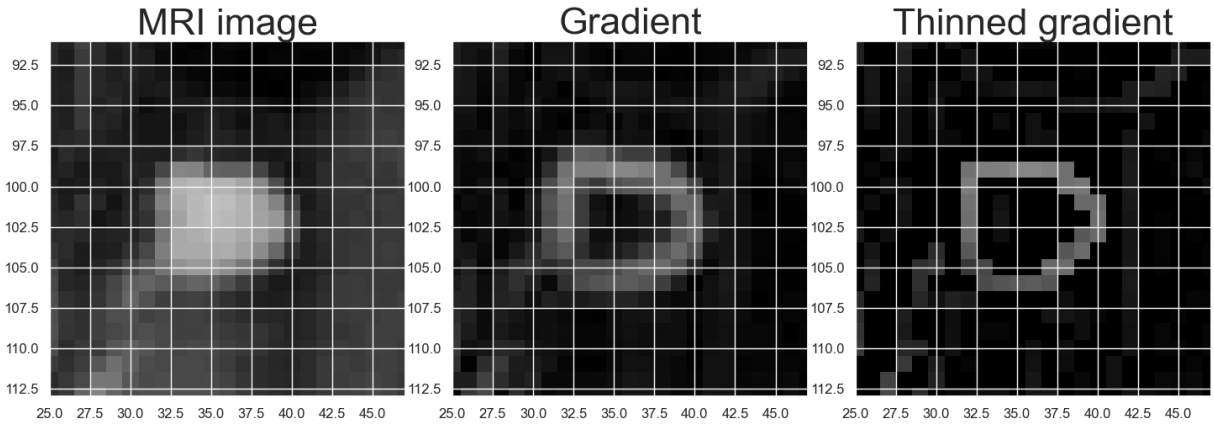

Figure 2.11: Plot of the gradient and thinned gradient for P1C at $t = 0.4$ s.

A high gradient indicates a large change of intensity between two neighbouring pixels and thus a border. The gradient magnitude and direction can be seen in Figure 2.12.

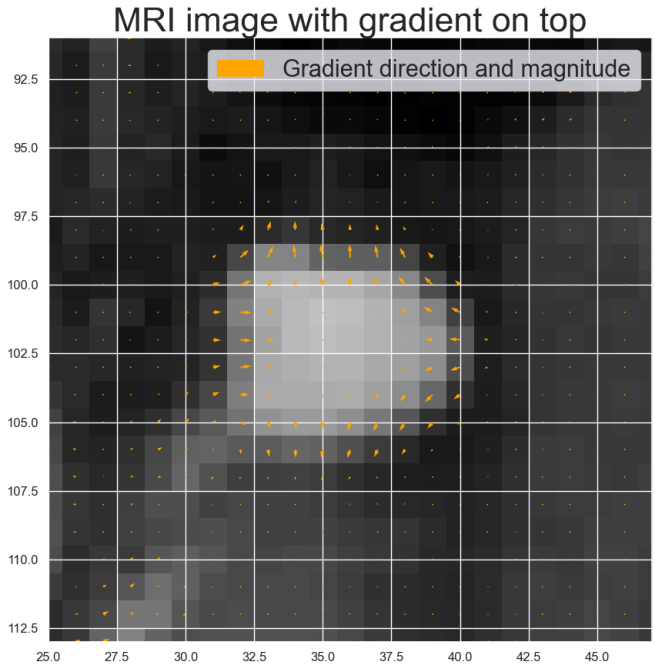

Figure 2.12: Representation of the gradient magnitude and direction on the MRI image of P1C at $t = 04$ s.

The next step is called "edge thinning". It allows to reduce the border thickness by taking the pixel with the highest gradient magnitude from a large border and taking into account the direction. In the implementation, the function takes the direction into account by scanning the pixels and storing the value of the pixels in the direction of the gradient. Thus, the algorithm allows to keep stored pixels with the maximum gradient value and the value of other pixels stored is put to 0. This makes it possible to refine the contour by keeping only one pixel on a contour that is wider as shown in Figure 2.11. A finer gradient is interesting because the algorithm can have more accurate results.

After that, the pixels that make up the gradient are separated in two classes. To do that, two thresholds are used. Pixels whose value is higher than the high threshold are classified as *strong borders* and pixels whose value is between both thresholds are named *weak borders*. The rest of the pixels are considered as background, i.e. their value are put to 0. The thresholds can be defined manually by trying different combinations of values and choosing the ones that give the best results. This is called the inverse parametrisation. However, this method is time-consuming because the values can be very different for each patient and there is no specific rules of proportionality to follow between the thresholds. Therefore, in order to not have to try too many low and high thresholds, we tried to automate the algorithm for each patient. We can opt for different strategies:

- One method is to define the thresholds proportionally to the maximum gradient value. Indeed, for each patient, the brightness level is very different and the gradient values are thus very different. This makes it possible to take into account differences between patients. However, we have to optimize the two proportionality coefficients. It is thus not the optimal strategy.

- The high threshold can also be found using the Otsu method as suggested by the article [17]. This method allows to find a threshold that separates well the different elements in the image in two parts thanks to the pixels intensity. The Otsu separation intensity can be taken as the strong threshold while the weak threshold would be proportional to it. As for the first method, an optimization step is needed to find the proportionality coefficient.

- We also tried another method which is more adapted to patients and views (sagittal or coronal). On the first five contours that we get from the physician, we take the mean number of pixels n that compose the contour. This number allows to define a good threshold. Indeed, by looking at the histogram of the enhanced gradient (cf. Figure 2.13), we obtain the pixel value for which there are n pixels above, i.e. pixels with a higher gradient value. This pixel intensity value is taken as strong threshold since we are sure there are at least n pixels with higher intensity changes above, we only keep the n most marked transitions of pixels. To determine the weak threshold, we can imagine keeping a safety margin of a certain number pixels that can be less bright but still belonging to the contour. This pixel number could be adjusted to the size of the tumor contour and thus to the brightness intensity of the gradient. This is why we define it as $1.2n$. It considers the intensity where there are $1.2n$ pixels above, as it is shown in Figure 2.13. This safety margin is necessary since the tumor can be deformed and the contour can consequently be larger and count more pixels. We have decided to focus on this last method.

All pixels classified as weak borders are then checked. If one of its neighbouring pixels (left, right, up, down) is already in the strong border category, the pixel becomes a strong border. If not, the pixel becomes part of the background. The result is therefore dependent on the order in which the pixels are checked. Let's take the following example: we have three pixels aligned: strong, weak and weak. If the algorithm starts from the left, the result is strong, strong and strong. Indeed, the first weak will become strong and the last weak will also have a strong neighbour. If the algorithm starts from the right, we obtain: strong, weak, weak.

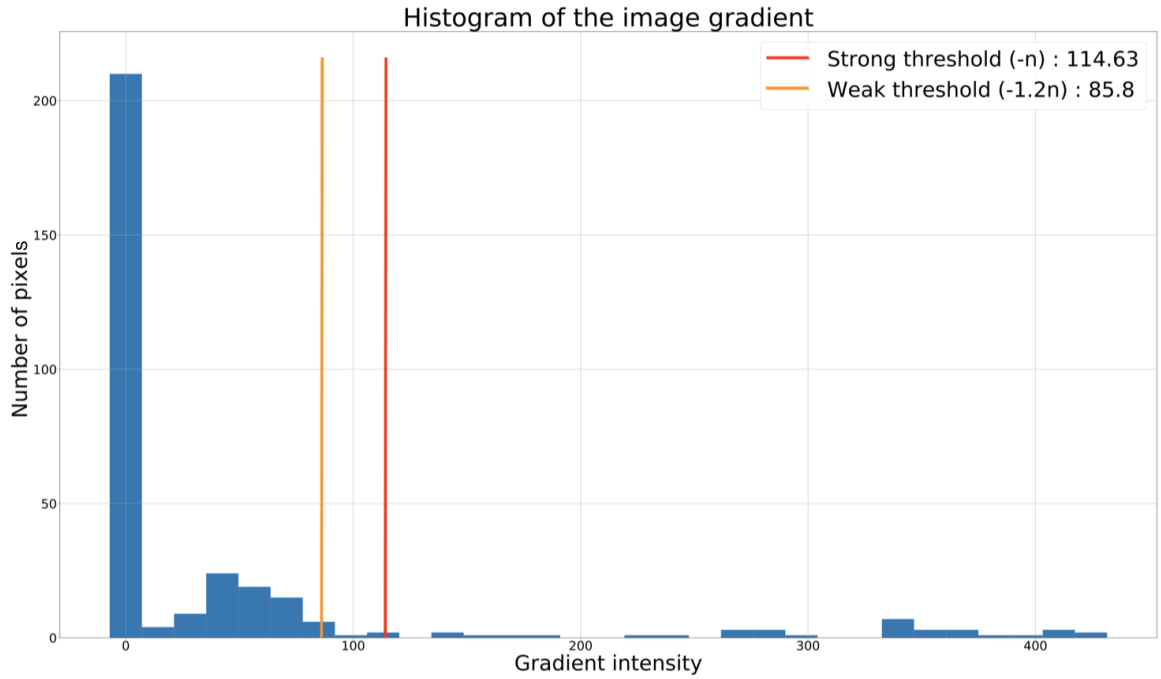

Figure 2.13: Histogram of the gradient MRI image of P1C at $t = 0.4$ s.

In Figure 2.14, the different steps of the Canny edge algorithm are represented. After cheking the weak borders, we see that some single strong pixels are left and are often not part of the contour. They are considered irrelevant since they have no neighbour belonging to the contour. We thus decide to apply a filter that eliminate the single pixels of the image. The filter puts the pixel value to 0 if all neighbouring pixels are black (0).

The contour is finally found by taking the remaining strong borders.

To take into account the fact that the physician's contour is always a bit bigger than the exact contour of the tumor, we again tried to dilate the contour. The same training set can be used to determine whether this gives better results. As for the Region growing algorithm, we used the training set to vary the parameter (i.e. the number of pixels in the contour) and determine which one gives the best results.

For the implementation, the majority of the code is based on the one found on [52].

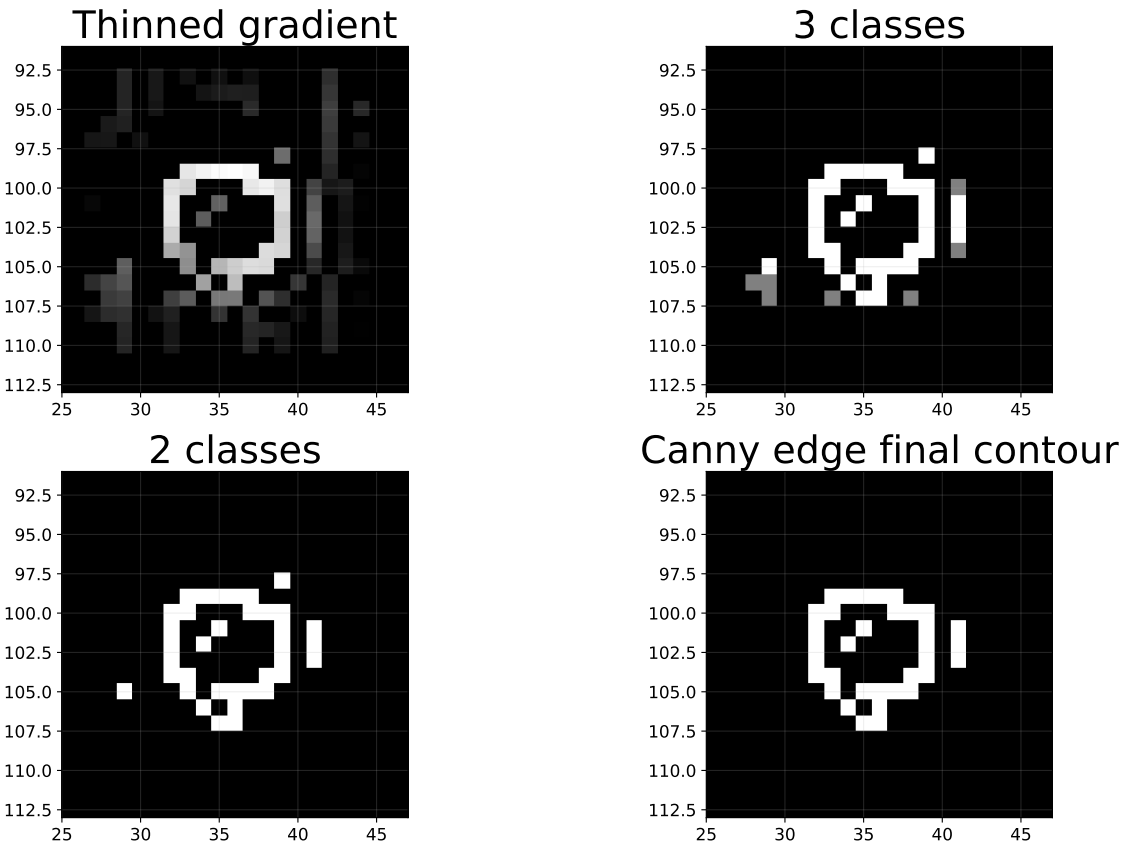

Figure 2.14: Steps of the Canny edge algorithm on P1C at $t = 0.4$ s.

3.3 Snake algorithm

The active contour model is a segmentation algorithm based on deformable models and is widely used since 1988 [53]. This kind of algorithm (here the snake) presents lots of variants like Gradient Vector Flow (GVF) snake model, the balloon model, the Diffusion snakes model and the Geometric active contours [54]. Those are used mainly in computer vision, object tracking, shape recognition, segmentation, edge detection and stereo matching [54]. The principle of the Snake algorithm is to find the contour of an element based on different energies. An initial contour composed of points of the image is first given as input. The precision of the initial contour depends on the application and at each iteration, its dynamics is governed by certain metrics. Those metrics are called here the energies. Indeed, the energies will guide the snake to the border of the tumor, so its contour. The algorithm can judge the quality of this first contour and adapt it to optimize those metrics iteratively. When those energies are optimized, the best contour is found. The different energies used can be defined differently following the problem to solve. The precise algorithm used and its different steps are explained just below.

The first step of the Snake algorithm is to crop the image to limit the algorithm to our region of interest, the liver, and even more restricted, a certain number of pixels around the tumor. This is done using the same method as the one used for the Canny edge algorithm.

The second step is to set a certain number of points representing the initial snake. This step is made easy with our application. Indeed, our project is to predict the position of the tumor at a certain time step. The prediction of its position consists in the prediction of 5 parameters which constitute an ellipse, (Y, X, a, b, α). This ellipse is then close to the real contour of the tumor if the prediction algorithm works well. The assumption is that the prediction is a good approximation of the tumor at this time step and therefore a good initialization of the algorithm.

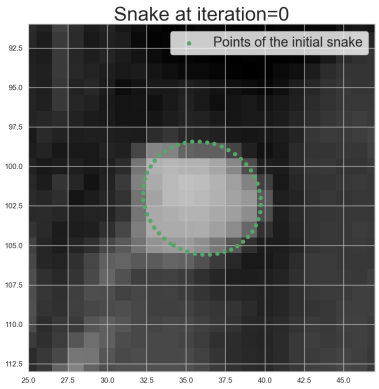

Figure 2.15: Initialization of the snake.

To have a better convergence, it is reasonable to take as first guess a reduction of this ellipse. Indeed, we can use the ellipse defined by $(Y, X, a-2, b-2, \alpha)$ instead of the one with half-axes a and b without reduction. By taking a smaller value of axes, and thus a smaller ellipse, it ensures the fact that the ellipse will be mainly comprised in the tumor. This technique ensures a faster convergence of the algorithm since it will be firstly attracted to the border of the tumor. Indeed, if the initial snake is outside the tumor, there is more chance that it could be attracted to the walls of the liver or other components in the organ. This will lead to more iterations of the algorithm. For our application, this is problematic since it has to be used during treatment and needs to be efficient and fast enough. This technique allows thus to limit the number of iterations and to speed up the convergence. The difference between both snake initializations can be seen in Figure 2.16 and Figure 2.17. We can observe that after the same number of iteration, the contour found by the algorithm seems more precise when the reduced ellipse is used as initial guess. We always take the approximated ellipse as initial snake at each time step as represented in Figure 2.15. The final goal is to really use the predicted ellipse to join the prediction and segmentation algorithms.

The number of points also influences the convergence. Indeed, more points lead to a more precise contour but need more time to converge and vice-versa. The number of points in the initial snake can be adapted to the mean size of the tumor. Indeed, as it is mentioned in Table 5.1, surfaces of the tumor differ from one patient to another. When the tumor is bigger, more points will be needed to have a precise contour. Here the number of points contained in the predicted ellipse is 100 and a certain proportion of it can be taken to be part of the snake. One third can be used for the first patient since the contour is composed of 36 pixels, while for the two other patients one half of the points are used in the snake since the tumor is bigger. The number of points in the snake does not vary from one iteration to the other, and the number is taken so that it is big enough to be accurate but not too big for the real-time application of our algorithm.

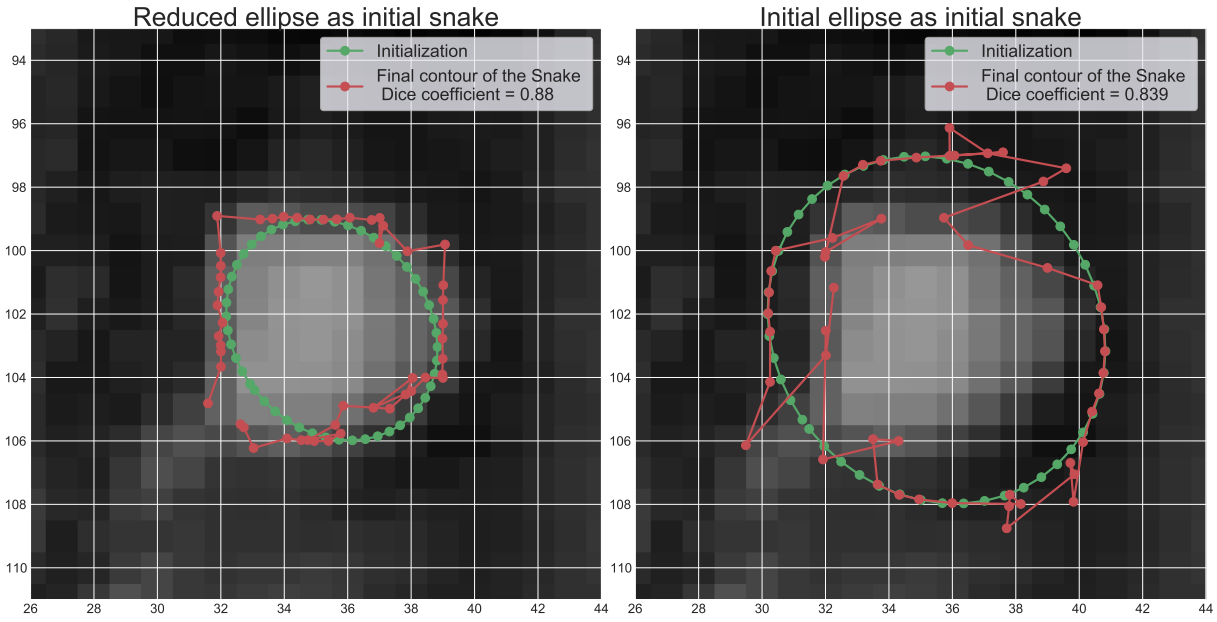

Figure 2.16: Initial and final snake for normal and reduced approximated ellipse for P1C at $t = 3$.

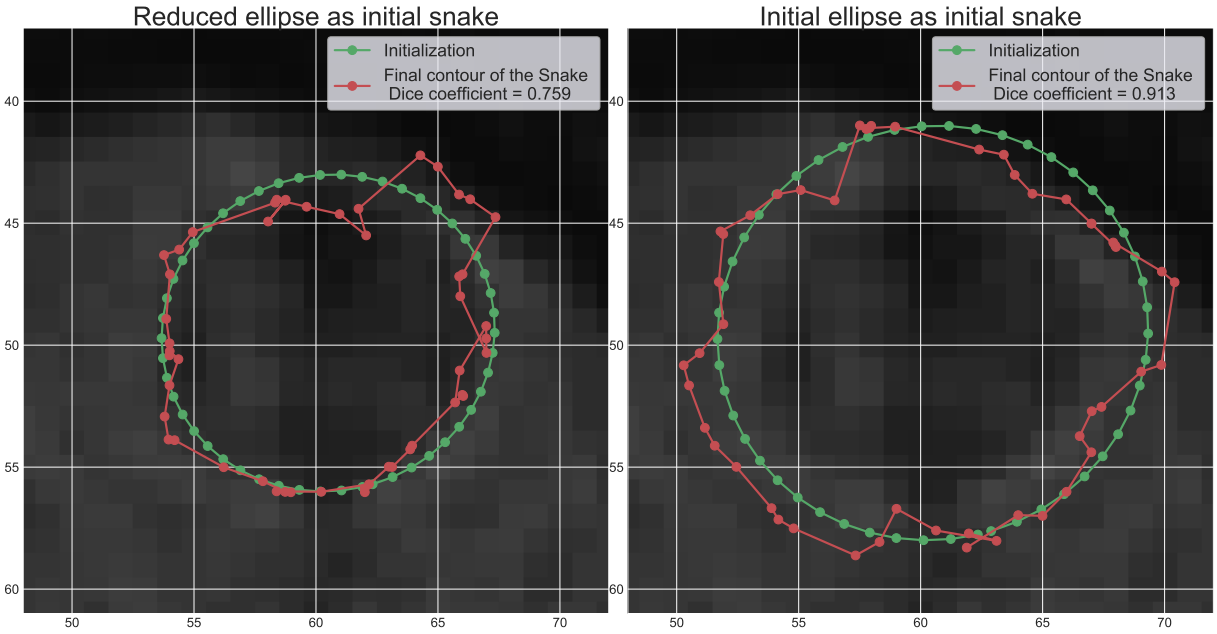

Figure 2.17: Initial and final snake for normal and reduced approximated ellipse for P3C at $t = 3$.

The goal of the third step is to create an image the algorithm can use to calculate certain forces. Based on these forces and the gradient, the contour can be orientated to find the perfect contour of the object. To do this, it is rewarding to use the gradient as it gives a lot

of information about sudden changes of intensity in the image. The external energy map (emap) is taken by converting the cartesian coordinates of the Sobel operator Gx and Gy to polar ones by taking the magnitude and angle of 2D vector. It is interesting to apply a thinning edge afterwards which highlights the edges of the components of the image. As previously stated, thinning will allow only the main part of the edge, highlighted by the gradient, to be retained so that only a specific edge is retained amongst a wider and potentially blurrier edge. Refer to the Section 3.2 of this Chapter to have more details about the functioning of edge thinning.

The fourth step is to define the metrics that we have introduced above. Indeed, the principle of the snake is that the final state that normally fits perfectly the contour of the liver tumor is a state of lower energy based on a defined energy function E_{snake}. To measure the snake energy, we can define different metrics that can favor some specific behaviours of the snake. A lower energy means thus that the current snake curve respects the imposed contraints. The final energy of the snake can be decomposed in two parts: an internal and external parts, that can also be decomposed, reinvented or refined following the problem. We can thus pose:

$$E_{snake} = E_{int} + E_{ext} \tag{2.24}$$

with

$$E_{int} = \alpha E_{cont} + \beta E_{curv} = \alpha |\frac{dv_i}{ds}|^2 + \beta |\frac{d^2 v_i}{ds^2}|^2 \tag{2.25}$$

and

$$E_{ext} = \gamma E_{image} = \gamma(-\nabla I(v_i)) \tag{2.26}$$

where

- E_{int} represents the internal energy of the snake while E_{ext} characterizes the external energy of the snake.
- The internal energy is composed of the continuity energy E_{cont} and the curvature energy E_{curv}. Their definitions are given by the equation (2.25) and their terms are developed in Equations (2.27) and (2.28) [53].

$$|\frac{dv_i}{ds}|^2 \equiv |v_i - v_{i-1}|^2 = ||x_i - x_{i-1}||^2 + ||y_i - y_{i-1}||^2 \tag{2.27}$$

$$|\frac{d^2 v_i}{ds^2}|^2 \equiv |v_{i-1} - 2v_i + v_{i+1}|^2 = ||x_{i-1} - 2x_i + x_{i+1}||^2 + ||y_{i-1} - 2y_i + y_{i+1}||^2 \tag{2.28}$$

  with x_i and y_i, the x- and y-coordinate of the point i of the snake.

- α, β and γ are parameters that guide the proportion of each energy in the final optimization.

The first component of the internal energy is the continuity energy. This ensures that the points of the snake are equidistant or not, and therefore allows the elasticity of the snake. To take this into account, the value of this energy is the sum of the squared distances between each point and its neighbour (cf. equation (2.27)). The curvature energy E_{curv} defines the fact that our snakes can have curved shapes or not. So it is directed by the

angles between three points of the ellipse. Indeed, acute angle between points defines a jerky shape. As given by the equation (2.28), it is defined by the second order derivative approximated by the finite differences.

The external energy is therefore composed of the energy of the image E_{image}. This energy contained in the image itself can be read on the emap. Indeed, the emap as described above, contains the necessary information about the edges and intensity changes in the image. This energy causes the snake to be drawn towards the edge of the tumor and the strong elements of the image. The energy is taken as the negative sum of the gradient of all the pixels that composed the snake in the image considered. If the value is small, this indicates that the snake might be at the right position. Indeed, contours are characterized by high gradient values. Based on the emap (cf. equation (2.26)), a high gradient means a low energy.

This is not an exhaustive list of the energies that can be used. Indeed, depending on the application, other constraints can be added. The perimeter, area or number of pixels contained in the contour can also be taken into account by adjusting the formula of some energies. For this work, only the energies presented above will be used.

Once all this is done, the different energies can be computed and optimized along the iterations and the algorithm can propagate from one slice to the other in time.

We also varied the different parameters to see their influence on the results. Theses parameters can also be optimized using the training set. The optimal parameters are the ones that maximize the Dice coefficient introduced in Chapter 4.

For the implementation, we first define the points that compose the snake and the energy map. After that, the different energies are computed following the definitions given. Once the total energy is computed, it is optimized using the Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithm. It is an iterative optimization method that determines the descent direction based on the gradient and curvature information and improves the loss function (or metric) (i.e. here the total energy) at each iteration. The optimization process stops when the minimum energy is found or when the maximum number of iterations is reached. The code for the optimization part is based on the one from [2].

3.4 Other segmentation algorithm

Watershed is an algorithm based on the gradient of the image considered as a relief map. In this one, one can consider the minima as the bottom and from these ones, they can make begin a stretch of water which will thus propagate and fill the holes. The objective of this algorithm is to find the watershed lines that correspond to barriers where the different water starting points cannot merge. Indeed, water from different minima will start to propagate after a certain time. The algorithm stops when all the peaks are underwater and only the barriers are visible from above. The different steps of the watershed algorithm are represented in Figure 2.18. If applied on the gradient instead of the image, the regions found should correspond to the homogeneous grayscale regions of this image [35].
This method can produce over-segmented results image since there can be noise in the initial image or irregularities. This can be avoided by different techniques such that

isolating some minima [35] or by the marker-controlled watershed. In this version, water begins to flood from a set of markers previously defined, instead of minima [9].

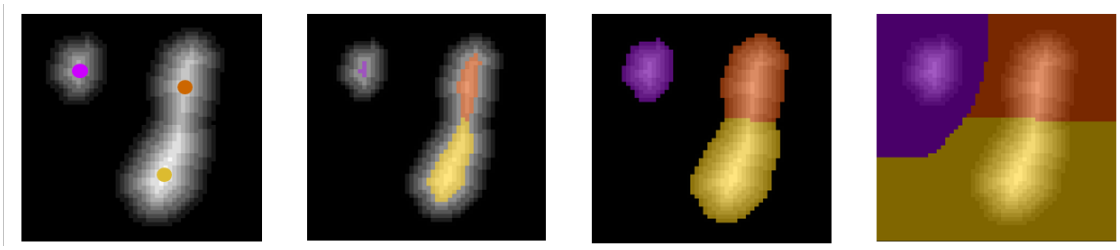

Figure 2.18: Flooding of a 2D image. On the first image, the minima are represented. The remaining images represent the propagation of the minima to fill the holes [35].

4 Combination of prediction and segmentation algorithms

The major goal of this application is to use simultaneously both algorithms of prediction and segmentation. The first one to predict the region where the tumor will be some time steps in advance and the other to know exactly the location of the tumor to be able to adjust the treatment in case of a bad prediction. Untill now, the segmentation algorithms were initialized thanks to the approximated ellipse obtained from the physician contour. This enables us to get the optimal parameters of our different algorithms based on an ellipse that was very well located. The objective is now to see if those segmentation algorithms work fine on predictions (coming from the prediction algorithms). This will thus not always be perfect located ellipse as before. As the performance of the different prediction models were tested, the best predictor is known. This last one is used to predict parameters of the ellipse on the testing set. Those predictions are got and given to the segmentation algorithms to see which one will be able to best recover the perfect surface of the tumor based on a prediction sometimes shifted.

Chapter 3

Material used

A Siemens Skyra 3T scanner allows us to get True FISP sequences. Abdominal compression and audio-coaching are used to try regularizing the breathing pattern. For each patient, the different positions are taken the same day and under the same conditions and are used independently. This means that each view even for the same patient is used as unique information over the movement of the tumor. The different views for a same patient do not add information to each other.

The MRI images are acquired with a frequency between 1.5 and 3.16 Hz. This is a compromise between the time needed for the acquisition and the quality of the image. The pixel size is thus of 1.82 mm. Different slice positions are given to study the tracking of the tumor motion:

- 2 interleaved coronal and sagittal slices (2 sequences for 4 positions) - 1.83 Hz for both orientations (3.66 Hz in total), 440 slices (2 x 220) in 2 minutes - TE: 1.39, TR: 3.16, 188x192 pixels of 1.82x1.82x7 mm
- 3 contiguous slices in sagittal or coronal plane (1 sequence for 3 position) - 1.5 Hz for the 3 slices (4.5 Hz in total), 540 slices (3 x 180) in 2 minutes - TE: 1.33, TR: 3.01, 126x128 pixels of 2.5x2.5x7 mm.

The data are composed of MRI image sequences from three different patients as summarized in Table 3.1. For each patient and each view, a file containing the MRI images and a file containing the tumor contours defined by the physician are acquired. The MRI images can be seen in Figures 3.1, 3.2, 3.3.

The language used for the code is Python and the SIPPY package is used for system identification [42]. For some codes, we were inspired by the following sources [19], [52] and [2].

P	V	Av. surface [mm^2]
1	C	278.42 ± 17.41
1	S	247.34 ± 24.52
2	S	633.68 ± 22.10
2	S	673.96 ± 18.84
2	S	593.53 ± 32.10
3	C	679.57 ± 59.80
3	S	706.04 ± 71.80

Table 3.1: Summary of the average surface and the standard deviation for all patients P and the corresponding views V. C corresponds to coronal and S to sagittal.

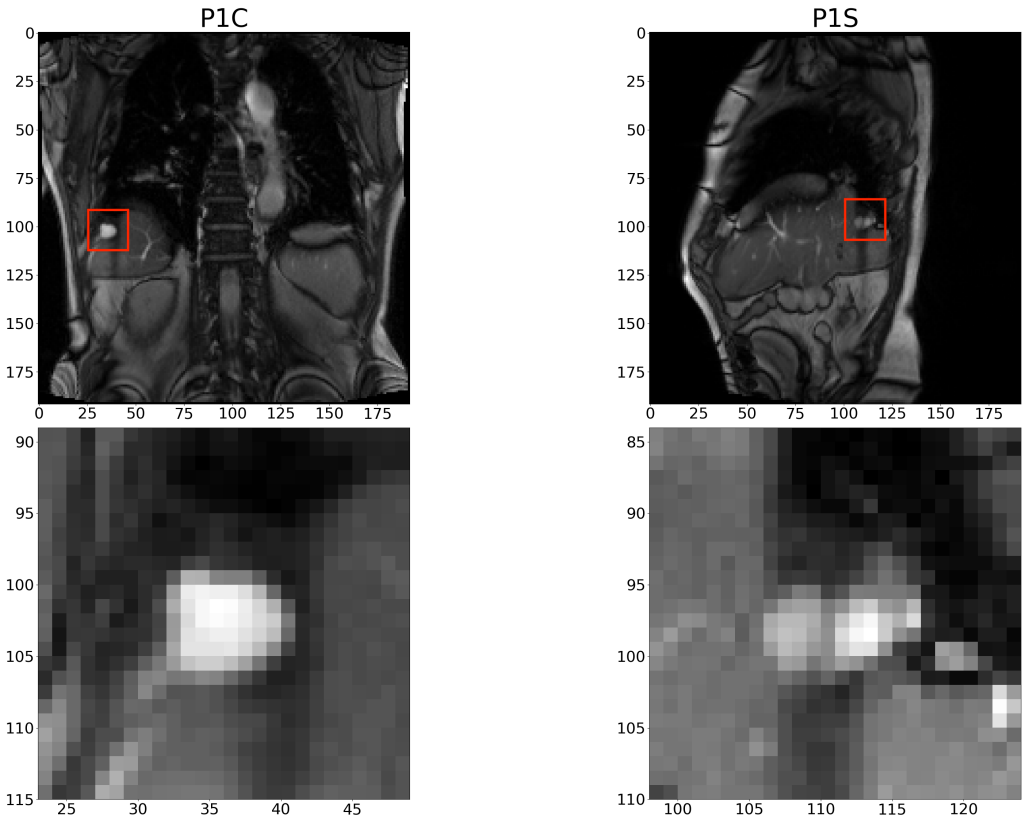

Figure 3.1: MRI images of P1 and MRI images zoomed on the tumor.

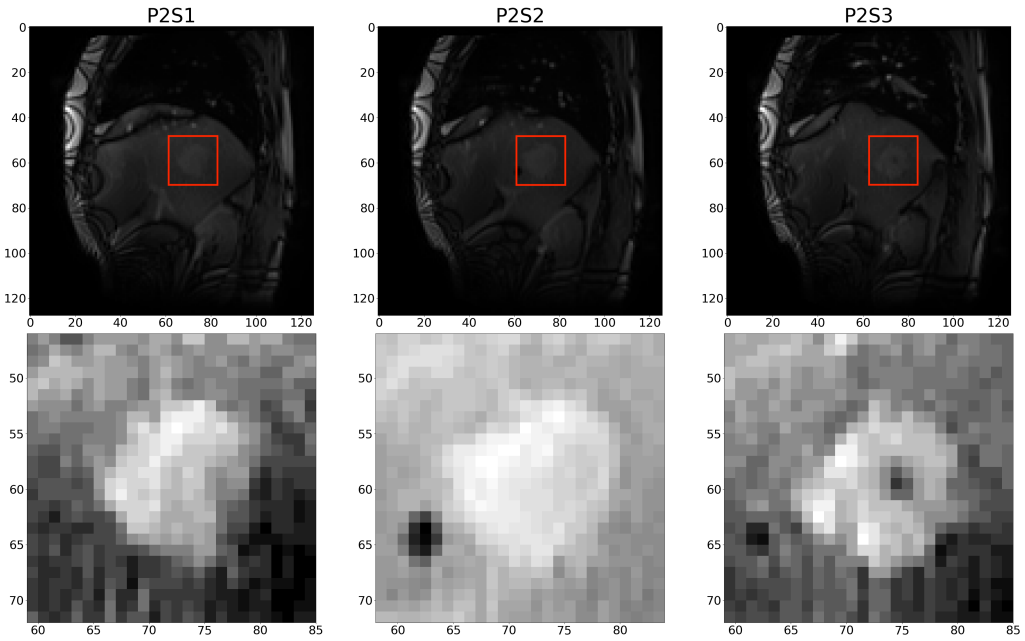

Figure 3.2: MRI images of P2 and MRI images zoomed on the tumor.

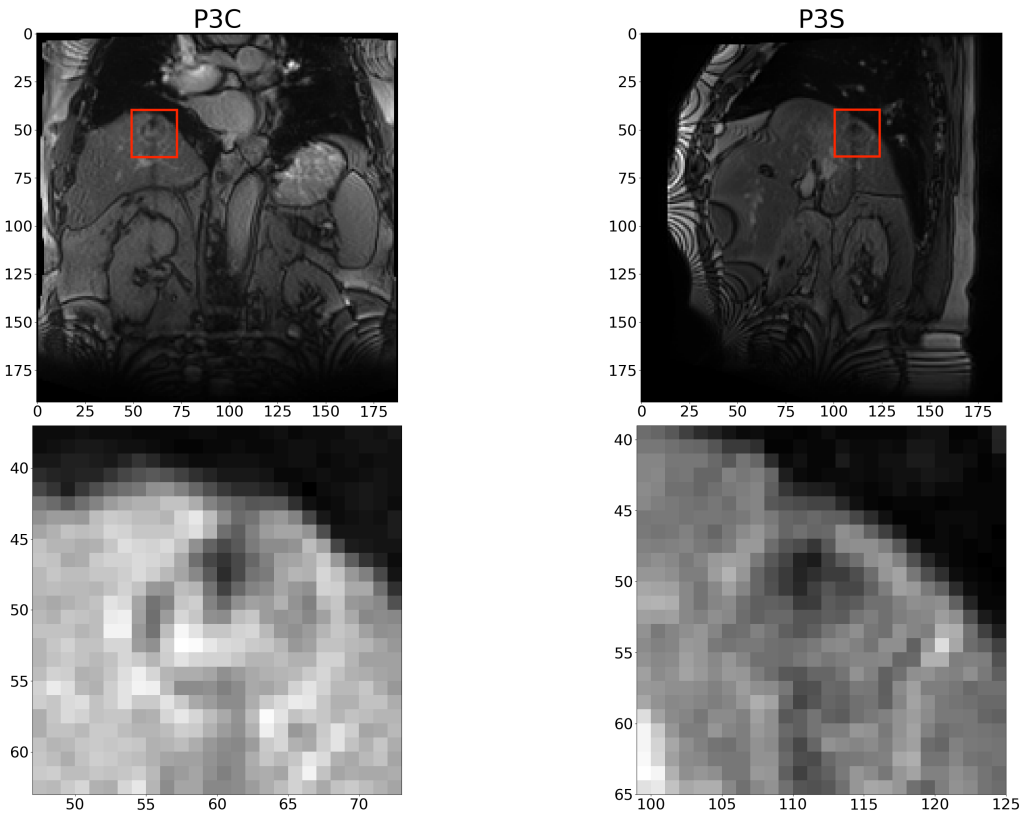

Figure 3.3: MRI images of P3 and MRI images zoomed on the tumor.

Chapter 4

Precision and evaluation

1 Baseline models

To evaluate the different models and algorithms, simpler models, namely the baseline models, are used. This allows us to compare the performance of our methods to a basic one. Those approaches are the most intuitive and are not expected to give the most performant results. We expect that our methods have better results.

1.1 Prediction models

The baseline model for the prediction models consists of considering the position and the shape of the tumor constant between two states k and $k+h$. The prediction for time step $k+h$ corresponds thus to the noisy measure of time step k. This means that

$$(\hat{z}_{1,k+h}, \hat{z}_{2,k+h}, \hat{z}_{3,k+h}, \hat{z}_{4,k+h}, \hat{z}_{5,k+h}) = (y_{1,k}, y_{2,k}, y_{3,k}, y_{4,k}, y_{5,k}) \tag{4.1}$$

The baseline model is represented in Figure 4.1.

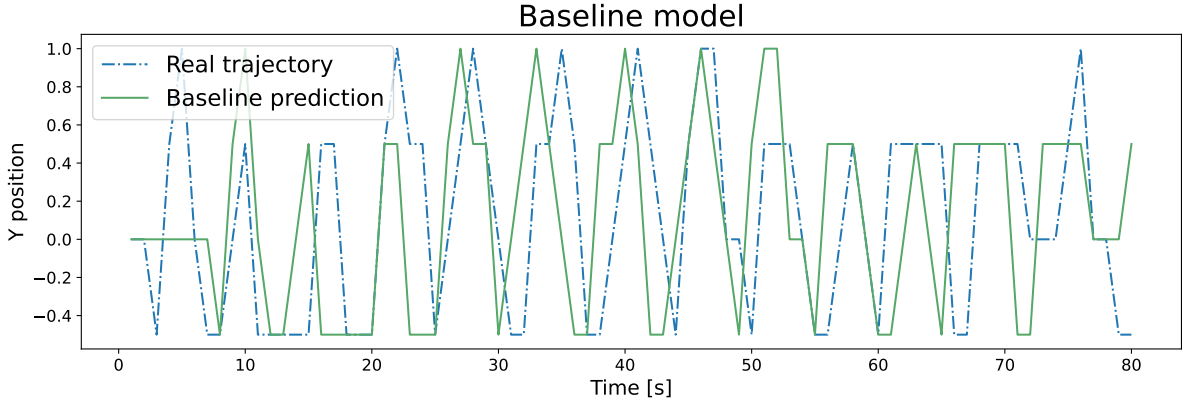

Figure 4.1: Baseline model with a shift h of 5 time steps.

The trajectory is thus the noisy measures shifted from h time steps. Even though it is a simple model, we could expect that it gives sufficient results on a small range of time.

Our prediction algorithms are firstly evaluated on their performances to predict only the tumor center of mass (i.e. $(z_1, z_2) = (Y, X)$ trajectory). The next step is to figure out

the results of the prediction of the 5 parameters $(z_1, z_2, z_3, z_4, z_5) = (Y, X, a, b, \alpha)$ which are used to rebuild an ellipse. The entire trajectory of the tumor surface is thus modeled; horizontal and vertical movement, stretching, shortening and rotation.
In addition, the prediction algorithms are tested on two types of trajectory, as it can be seen in Figure 4.2:

- On a perfect sinusoidal trajectory that represents a regular breathing. The regular breathing pattern for the Y and X positions are modelled by : $y_{regular} = 2\sin(4.57t)$, $x_{regular} = -0.8\sin(4.8t)$. The different values are chosen based on the real trajectory. Indeed, the period and the amplitude are calculated on the x and y trajectories of P1C. We would expect the same period for both sinus as it is determined by the respiratory cycle. The periods are different because they are calculated independently of each other and influenced by irregularities.
- On the real trajectory $z_i(t)$ that represents a real and irregular beathing.

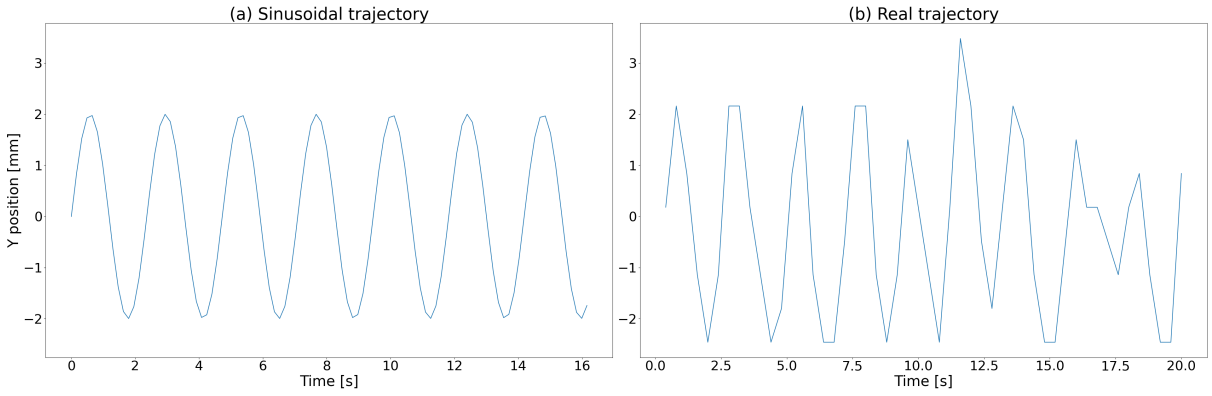

Figure 4.2: Representation of both types of trajectory: (a) Regular breathing pattern. (b) Irregular breathing pattern.

1.2 Segmentation algorithms

The baseline model for the contour algorithms is simply given by the contour of the approximated or predicted ellipse without refinement which is represented by the green ellipse in Figure 2.2. This approach allows us to see the real impact of the refinement from our contour algorithms.

2 Error quantification

We use the physician contour as ground truth to be able to quantify the prediction. In practice, these contours are only needed when constructing the model on the training set. To quantify the error between the real $(z_{1,k}, z_{2,k}, z_{3,k}, z_{4,k}, z_{5,k})$ and predicted trajectory $(\hat{z}_{1,k}, \hat{z}_{2,k}, \hat{z}_{3,k}, \hat{z}_{4,k}, \hat{z}_{5,k})$, we first wanted to use the Root Mean Squared Error (RMSE) given by equation (4.2).

$$RMSE_i = \sqrt{\frac{1}{N}\sum_{n=1}^{N}(\hat{z}_{i,n} - z_{i,n})^2} \tag{4.2}$$

with N, the number of time steps on which the RMSE is computed.

As it will be mentioned in Chapter 5, the errors are always bigger when the movement amplitude is higher. To determine whether it is related to our models or to the dependence between the RMSE and the amplitude of the trajectory, we tried to use the Percent Relative Error (PRE) whose formula is stated in equation (4.3):

$$PRE_i = \frac{100}{N} \sum_{n=1}^{N} \frac{(z_{i,n} - \hat{z}_{i,n})}{z_{i,n}} \tag{4.3}$$

There is no dependence to amplitude anymore because of the division by the noisyless measure $z_{i,n}$ and thus the amplitude of movement at each time step. However, the real value can sometimes be equal to zero. Because of the division by zero, a problem is raised. That is why we decided to use the Symmetric Mean Absolute Percentage Error (SMAPE). The definition is given by the equation (4.4).

$$SMAPE_i = \frac{100}{N} \sum_{n=1}^{N} \frac{|\hat{z}_{i,n} - z_{i,n}|}{|\hat{z}_{i,n}| + |z_{i,n}|} \tag{4.4}$$

Thanks to the division by a sum of absolute amplitudes of movement, there is no problem of division by zero anymore. This division is made for all time steps where we have a prediction and the mean is then taken. This formula ensures that the SMAPE has an upper and lower bound since it will be comprised between 0 and 100%. It must be known that over- and under-forecasting are not treated equally. Indeed, if the predicted value $\hat{z}_{i,n}$ is greater than $z_{i,n}$ from a certain number, the SMAPE will be different than for a predicted value which is smaller than $z_{i,n}$ from the same number. The higher influence is given to the underprediction since this will lead to a smaller denominator.

To evaluate the performance of the identification stage of the Kalman filter prediction method (the system identification), the Explained Variance (EV) could be used (cf. equation (4.5)) [3]. However, we decide to mainly use RMSE and SMAPE.

$$EV_i = 1 - \frac{\sum_n (\hat{\epsilon}_n^{(i)})^2}{\sum_n (z_n^{(i)} - \bar{z}^{(i)})^2} \tag{4.5}$$

with EV_i representing the explained variance of the ellipse parameter i and n is the time step.

3 Dice coefficient

A percentage of recovery between two surfaces can be used as a measurement metric to determine whether:

- The ellipse approximates well the tumor contour.
- The ellipse obtained with the prediction models is a good prediction of the real contour.
- The contour found with segmentation algorithms is correct and performant.

The Sorensen-Dice coefficient is a metric that gives the recovery percentage. For two finite sets X and Y, the Sorensen-Dice coefficient $Dice(X, Y)$ is defined as

$$Dice(X, Y) = \frac{2 \mid X \cap Y \mid}{\mid X \mid\mid Y \mid} \tag{4.6}$$

with $|X|$, $|Y|$ the number of elements in the set X and Y respectively, $|X \cap Y|$ the number of elements belonging to the intersection between both sets.
The value of $Dice(X, Y)$ is comprised between 0 and 1 meaning the sets are respectively disjointed and equal.

To know whether the ellipse is a good approximation of the tumor contour, the Sorensen-Dice coefficent is used. To select the pixels belonging to the ellipse, a binary mask is first constructed. The pixels where the ellipse contour is situated are put to 1 and the rest remains black. After that, the shape is filled. These steps are shown in Figure 4.3. The ellipse is a good approximation if the Sorensen-Dice coefficient is near 1.

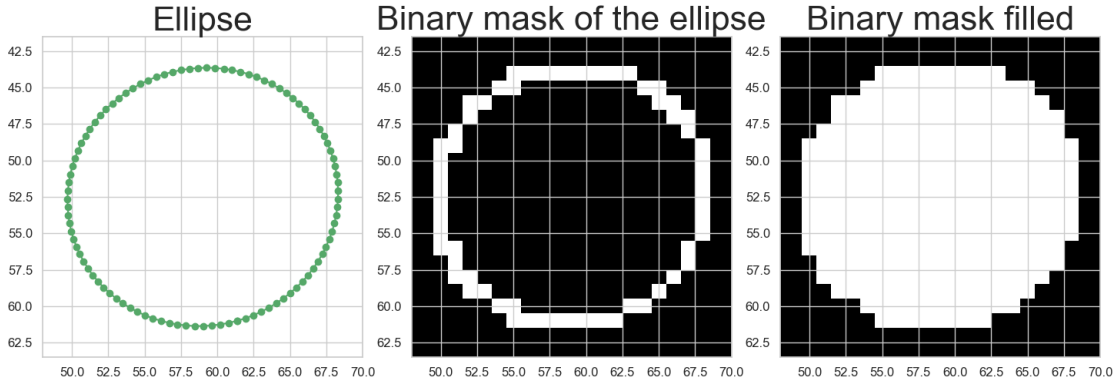

Figure 4.3: Steps to select pixels belonging to the ellipse. P3C at $t = 0.4$ s.

To have an idea of which performance can be qualified as good one, different articles were found. In the study [55], a multi-scale template matching algorithm is used to segment lung tumors on PET-CT images and the average Dice coefficient is around 0.858. As the soft tissue contrast is better on MRI images, we expect slightly better results. Indeed, Dice coefficients above 0.93 were reported when contouring a lung tumor on dynamic MRI images [56].
In another article, three different algorithms (Chan Vese, Otsu adaptive threshold segmentation and Region growing) were used to segment lung tumors on MRI images. The performance was measured with the Dice coefficient between the region found with the algorithms and the one contoured by experts. The results show that this value was equal to 0.860, 0.612 and 0.730 for respectively the Chan Vese, the Otsu segmentation and Region growing [16]. These are numbers we could take as reference.
A study has shown results about tumor segmentation in the brain based on MRI images. Thanks to their data set, they manage to test their algorithm with different loss functions; CE-based and metric sensitive ones. In general, the Dice coefficients have a mean value of 0.8-0.85, going from 0.808 to 0.871 [15].
In [6], based on MRI images, baseline models have been created to predict the progression of nasopharyngeal carcinoma. They evaluate the performance of their radiomic model

with the Dice Similarity Coefficient (DSC) between segmented regions of interest of each patient. They indicate a scale of goodness of this metric so that if it is lower than 0.6, it indicates inadequate consistency, while between 0.6–0.8, 0.8–1.0, and 1.0 it represents good, very good and ideal consistency.
Based on these studies, we can say that the performances are satisfying if the Dice coefficient is above 0.8.

For the prediction models, the sets are composed of pixels contained in the real contour, based on the MRI image of the predicted time, and the predicted ellipse.

The Sorensen-Dice coefficient is also used to quantify the performance of the segmentation algorithms. The sets used contain the pixels of the real contour and the one found with one of the algorithms.

The Dice coefficient is also useful to quantify the performance of our algorithms because when treating a tumor, knowing its surface is important as all tumor cells must be treated. As the surface is too large, healthy cells will be over-dosed and as it is too small, cancer cells will be under-dosed.

A simple and visual tool to represent the performances and to compare the different models in function of a parameter is the use of boxplots. An example of boxplot is represented in Figure 4.4. The yellow line is positioned at the median, the red rectangle includes the 25th percentile until the 75th and the blue segments go from the first to the ninth deciles. The points represent the outliers, they are determined by a method based on the inter-quartile range.

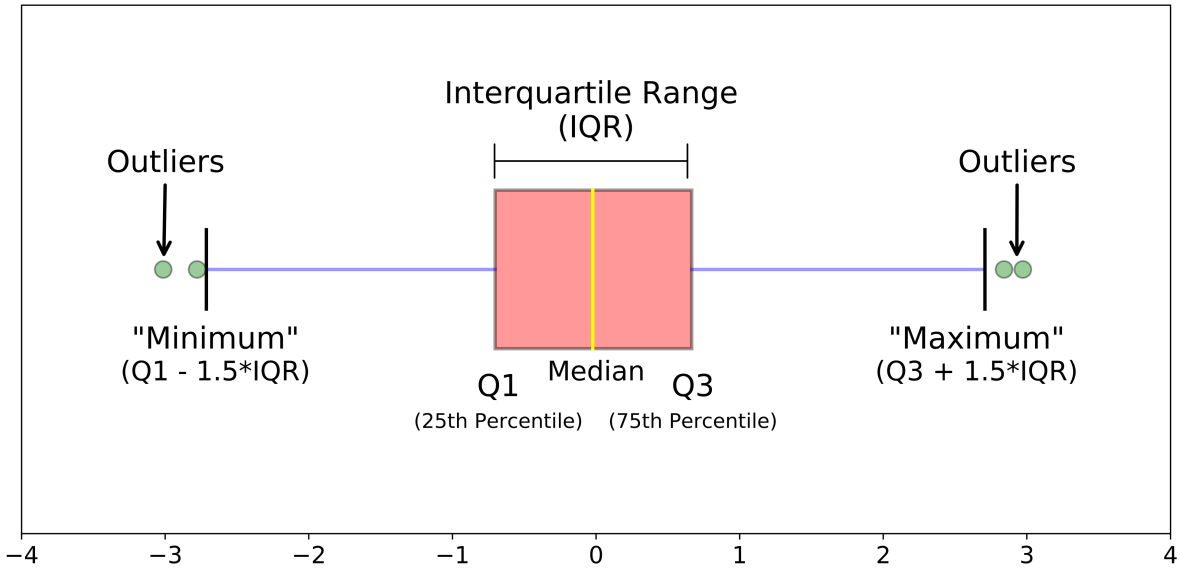

Figure 4.4: Boxplot representation [21].

4 Noise influence

Since we are dealing with imaging data, noise is always present. The goal is to be sure that our algorithms are robust to noise. The noise added to data are simulated randomly as explained in Section 1 of Chapter 2. To test the robustness, one can imagine to do several simulations of the same prediction models with different random noises added to raw datas $(z_1, z_2, z_3, z_4, z_5)$. The concerned models are the baseline, the sinusoidal models and the kalman filter since it is based on the system identification response. All those algorithms depend on the noisy measures. A set of 100 simulations will thus be performed for the training and identification stages while the validation and the testing stages will be evaluated on 50 and 100 simulations respectively. The validation stage is longer since it adjusts the hyperparameters and lots of different parameters have thus to be tested thanks to inverse parametrisation. This is why less simulations are performed.

5 Precision gain

As explained in Section 3 of Chapter 1, one of the most used solution to take into account tumor motion is to use an ITV-based PTV margins. To determine the ITV, the primary macroscopic tumor surface (the physician's contour) is delineated on each image of the MRI image sequence and the surfaces found are combined [29], [7]. In the article [7], isotropic margins of 5 mm, 4 mm and 3 mm were added to the ITV to take into account remaining uncertainties. To add a margin, we dilate the ITV obtained with a 5×5 kernel. The surface is increased by two pixels, corresponding to 3.64 mm. The ITV (in orange) and PTV (in red) obtained for P1C and P1S are shown in Figure 4.5. The white represents the physician contour, the orange the ITV and the red the PTV.

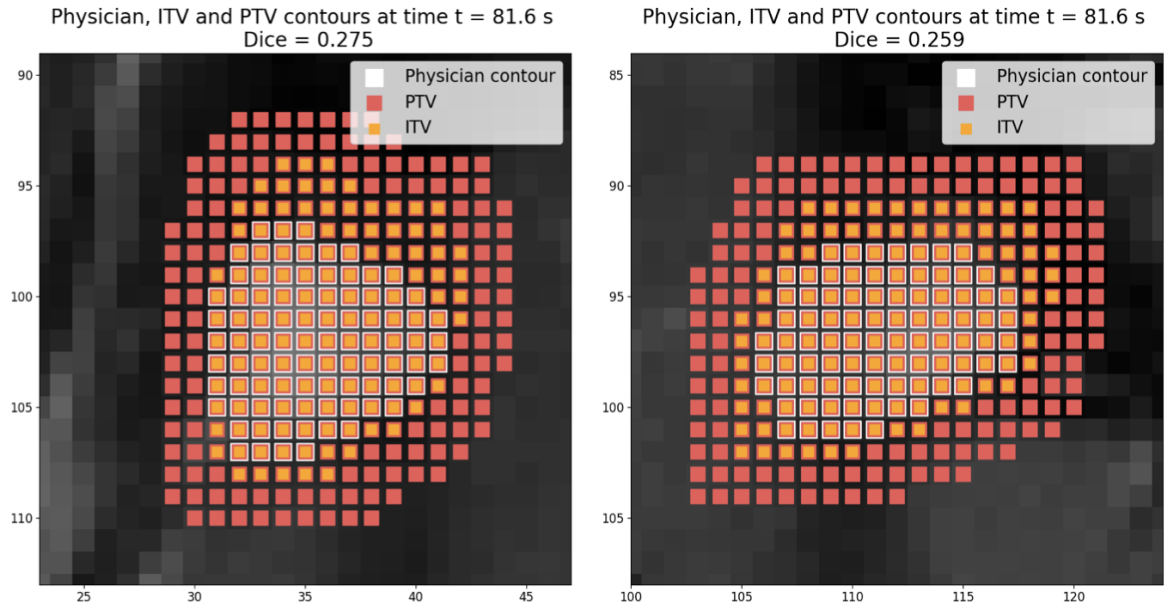

Figure 4.5: Physician contour (in white), ITV (in orange) and PTV (in red). Left: P1C, Right: P1S at $t = 81.6$ s

One method to evaluate the performances of the prediction models is to compute what is gained in terms of dose compared to the margins, what quantities have gone erroneously to

the patient with one method and the other. This gain is based on the surface of healthy cells and the proportion of tumor surface that are irradiated. By determining and comparing the surface and the proportion for both methods, margin-based and prediction-based methods, we can evaluate the gain. When evaluating, the proportion of the tumor not irradiated has a higher importance as it is important for the treatment that all cancer cells are irradiated. To find the surface irradiated or not, the number of pixels is used knowing that the pixel surface is 1.82×1.82 mm^2.

6 Real-time

To determine whether the different algorithms can be used in real-time, the time taken to run the code can be computed using the *time* funtion of Python. However, due to lack of time, this will not be evaluated in this work.

In our approach, two types of algorithm must be applied simultaneously, named a prediction and a segmentation algorithms. To be optimized, these algorithms must be applied in parallel. Indeed, the prediction algorithm allows to predict the parameters of time $t + h$. In parallel, the information given by image obtained at this time t is compared to the prediction of this time step (this prediction was made at time $t - h$ for time t) in order to determine the quality of the prediction and the treatment. Once the prediction is done for time $t + h$, a segmentation algorithm is applied and the information is retained until the image corresponding to this time step is obtained to perform the comparison. Then, when a new MRI image comes in, it is supposed to begin directly the segmentation algorithm since the prediction is already made for this time step. The ideal solution must be implemented thanks to Parallel computing which allows several tasks at the same time. This should make those algorithms partially independent and able to run simultaneously by using the data from each other. A schematic view of our entire application can be seen in Figure 4.6.

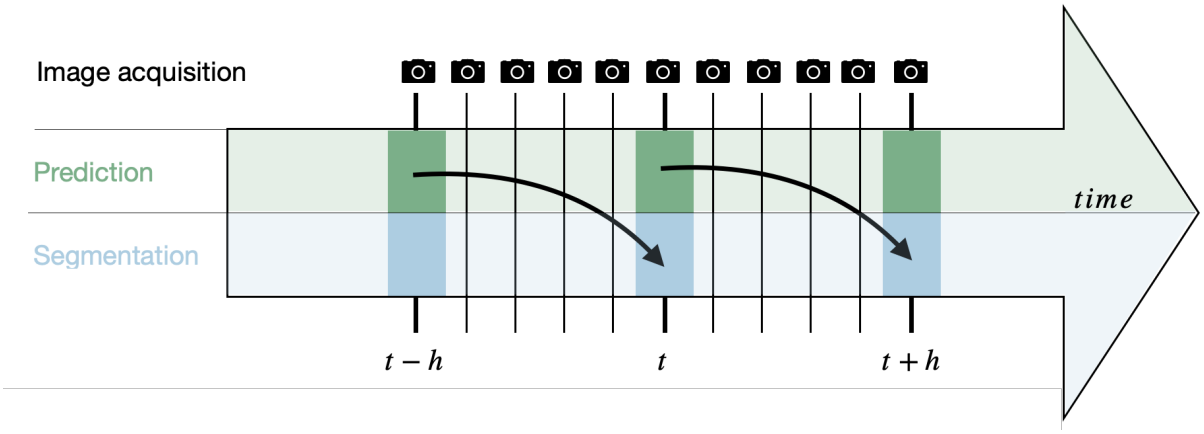

Figure 4.6: Timeline representing real-time operations

Based on theses steps and because the time interval between two image acquisitions is 400 ms, the execution time should be around 400 ms for both prediction and segmentation algorithms to be considered as real-time.

It all depends on the goal we give to this application and the time step to which we want to predict. Indeed, a prediction of one time step or two time steps (here 0.4 or 0.8 s) will not be sufficient for a physician to have any control while a prediction of 5 times steps (here 2 s) can lead to a fast decision of a physician to adapt the treatment following the prediction made. Moreover, following the prediction algorithm chosen (the Kalman filter for example), the performance is sometimes not the same in the 20-30 seconds than after 1 minute. This will be seen in the Chapter 5 since the algorithm needs a certain time of convergence and gives sometimes less precise results after a certain period of time.

Chapter 5

Results and discussion

All the figures depicted in this chapter are interpreted and then discussed in the black boxes. The limitations of our algorithms and the propositions of amelioration are stated in Chapter 6. The main conclusions are finally taken in Chapter 7.

1 Ellipse approximation

As mentioned in Section 1 of Chapter 2, two methods can be used to compute the a axis of the ellipse: PCA and most distant point. The Figures 5.1 and 5.2 allow to conclude about the best technique to approximate the tumor with an ellipse in a visual way for P1C and P1S, and in a quantitative way thanks to the Dice coefficient for each patient. This states the overlapping of the physician contour and the ellipse found. P1C and P1S are chosen because the tumors have two different shapes: one is round (coronal view) and one is more oval (sagittal view). The best technique is taken for the rest of the work.

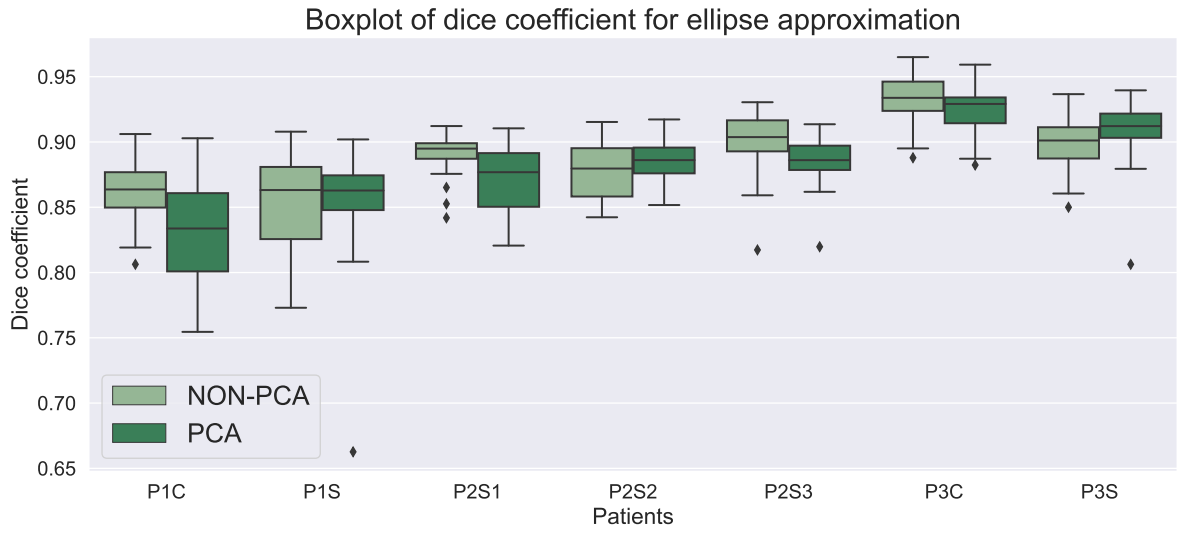

Figure 5.1: Boxplot of the Dice coefficient between the real contour and the approximation of the tumor with an ellipse. Two methods of approximation are used: PCA and most distant point (NON-PCA). All patients. Testing set.

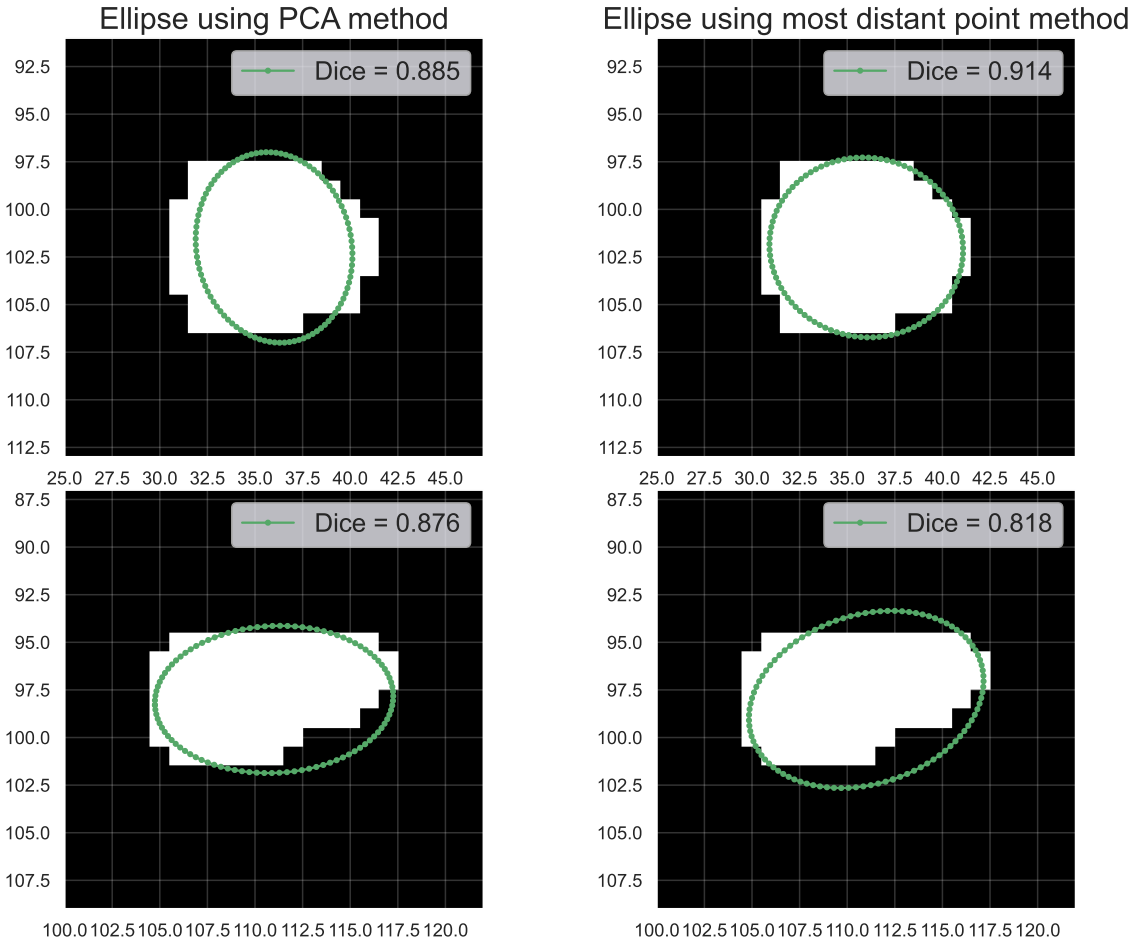

Figure 5.2: Comparison between two ellipse approximation methods. The approximated ellipse is represented in green and the filled physician contour in white. Left: PCA method. Right: Most distant point method. Up: P1C at $t = 0.4$ s. Down: P1S at $t = 0.4$ s.

We can observe in Figure 5.1 that the mean of the Dice coefficient for both methods is between 0.83 and 0.93. Moreover, their standard deviation is in the same range for both methods. The performances of both methods are thus quite similar but the Dice coefficient is a little bit higher for five out of seven patients with the most distant point methods. As it can be seen in Figure 5.2, a lower proportion of the tumor surface is included in the ellipse with PCA method than with the most distant point method for both patients. This arrives even if the Dice is better with this method for one patient.

We can conclude that the tumor is well approximated as the mean Dice coefficient is high (> 0.83) for all patients.
The higher tumor coverage of the most distant point method could be explained by the fact that the most distant point is always included in the ellipse. For our application, it is better that the whole tumor surface is contained within the ellipse as all cancer cells must be irradiated so that the treatment works properly. This is a disadvantage of the Dice coefficient: the penality is the same whether the overlap between the two sets is larger or smaller. For our application, the penality, i.e. the reduction of the Dice coefficient, should be lower in the case of higher overlap.

Based on these observations, we decided to use the most distant point method for the rest of this work.

After obtaining the five ellipse parameters at each time step, the trajectory of each parameter for P1C is traced in Figure 5.3. The trajectories for the remaining patients can be seen in Figures A.1, A.2, A.3, A.4, A.5 and A.6. This allows to obtain a global view of the movement of the seven tumors studied.

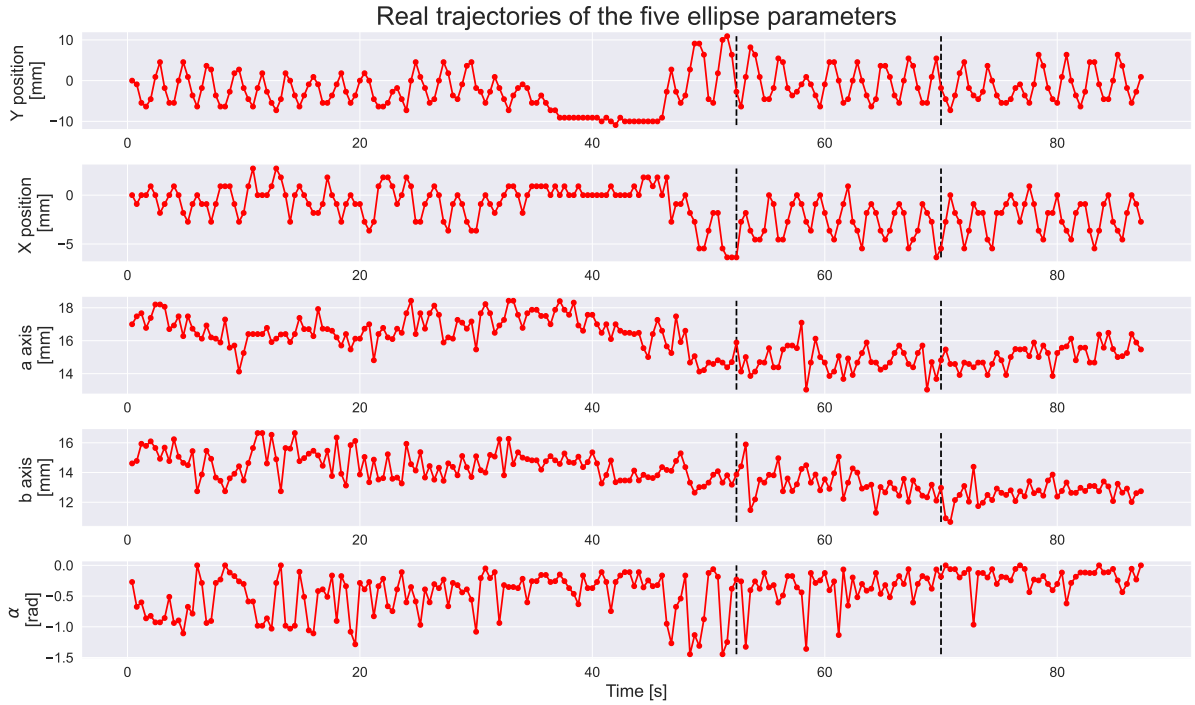

Figure 5.3: Trajectories for the five parameters of the ellipse (Y, X, a, b, α). The vertical lines represent the separation between the training, validation and testing sets. P1C. All sets.

Based on these trajectories, the amplitude of the five parameters trajectory is calculated and summarized in Table 5.1. This enables to understand whether the tumor is moving a lot or not. Knowing the mean surface (cf. Table 3.1) is also useful for the initialisation of the segmentation algorithms (as the number of points for the Snake algorithm).

		1	*2*	*3*	*4*	*5*
P	**V**	**Y amp. [mm]**	**X amp. [mm]**	**a amp. [mm]**	**b amp. [mm]**	**α amp. [rad]**
1	C	2.73 ± 0.91	1.09 ± 0.36	0.91 ± 0.45	1.04 ± 0.56	0.13 ± 0.016
1	S	2.46 ± 0.82	1.64 ± 0.55	1.10 ± 0.55	1.26 ± 0.94	0.20 ± 0.09
2	S	1.64 ± 1.23	0.55 ± 0.41	1.07 ± 0.8	0.77 ± 0.58	0.26 ± 0.08
2	S	2.05 ± 1.02	0.55 ± 0.41	0.96 ± 0.32	0.83 ± 0.28	0.19 ± 0.09
2	S	1.64 ± 1.23	0.55 ± 0.64	0.76 ± 0.57	0.55 ± 0.64	0.25 ± 0.19
3	C	3.25 ± 1.62	1.64 ± 1.23	1.12 ± 0.84	0.83 ± 0.63	0.75 ± 0.09
3	S	3.60 ± 1.20	2.73 ± 0.91	1.34 ± 1.01	1.71 ± 0.57	0.71 ± 0.09

Table 5.1: Summary of mean amplitude and standard deviation values of the ellipse parameters (Y, X, a, b, α) for the patients P and the corresponding views V. C corresponds to coronal and S to sagittal.

The major observation done in Figure 5.3 is that the trajectories of the Y and X positions have a more sinusoidal tendency and thus seem to be the most regular. In addition, the amplitude of the Y position is higher than the one of the X position. As it can be seen in Table 5.1, this observation can be done for all patients.
Moreover, we observe that the values of a and b axes are close to each other for P1C.
Another analysis is that the trajectory of the angle of rotation α shows less variations. In degrees, the variation is less than 90°.

The sinusoidal character of the Y and X position explains why we consider a sinusoidal model for the prediction. This is relevant since it is due to respiratory motion.
Based on the values of a and b axes, we can conclude that they are well defined. Indeed, their equality is explained by the round shaped tumor of P1C.
The small variations in α angle leads to the conclusion that the tumor does not rotate much between two time steps as stated in the assumptions of Section 1.

Concerning the Table 5.1, the amplitudes found correspond to what we can find in the litterature. For tumors closer to the diaphragm, the vertical displacement can go to 15 mm and the horizontal one is around 3 mm [29, 31]. The standard deviation found on the real trajectory can give an idea about the irregularities that appear in the trajectory. One can state that the Y position has more irregularities while for a, b or α, it cannot be said that there are many irregularities. Indeed, the variation of these parameters is not regular enough on the whole time studied to qualify some changes as irregularities. It remains thus complex to draw a conclusion on the irregularity of the signals.

2 Prediction models

Each prediction model is analysed independently. Based on their individual characteristics, prediction precision, advantages and inconvenients, the best one can determined.

In the figures of this section, the real noisyless trajectory is shown in continuous red line while the noisy trajectory (measurements) is located mainly in the lighter red region. Indeed, as we want to evaluate the robustness of the algorithm, we simulate the entire noisy trajectory 100 times, that is with 100 different noises for each time step. It is thus impossible to represent all the measurements on a graph. The standard deviation is thus filled in red to represent the region through which the noisy trajectories pass.
The prediction is shown in the same way in the color associated to the model: the mean of all predictions based on the different simulated noisy trajectories is represented in continuous line and the region in which all the predictions for all the simulations are mainly located in a ligther color. This concept of zone represented by the standard deviation of all the different signals due to stochasticity of the measurements are used for all methods.

The vertical line, which corresponds to the fifteenth time step of the set used, represents the moment at which the RMSE, the SMAPE and the Dice begin to be calculated.

2.1 Baseline model

The Figures 5.4 and 5.5 represent the performance of the baseline model on a regular breathing pattern and on the real trajectory (for P1C) on the testing test. The testing set is directly used since no model has to be built and no parameter has to be optimized. It allows to evaluate the performance of this model in a realistic situation in a visual and in a quantitative way thanks to the RMSE, the SMAPE and the Dice coefficent between the ellipse predicted (thanks to the 5 predicted parameters (Y, X, a, b, α)) and the tumor contour of the physician.

2.1.1 Regular breathing pattern

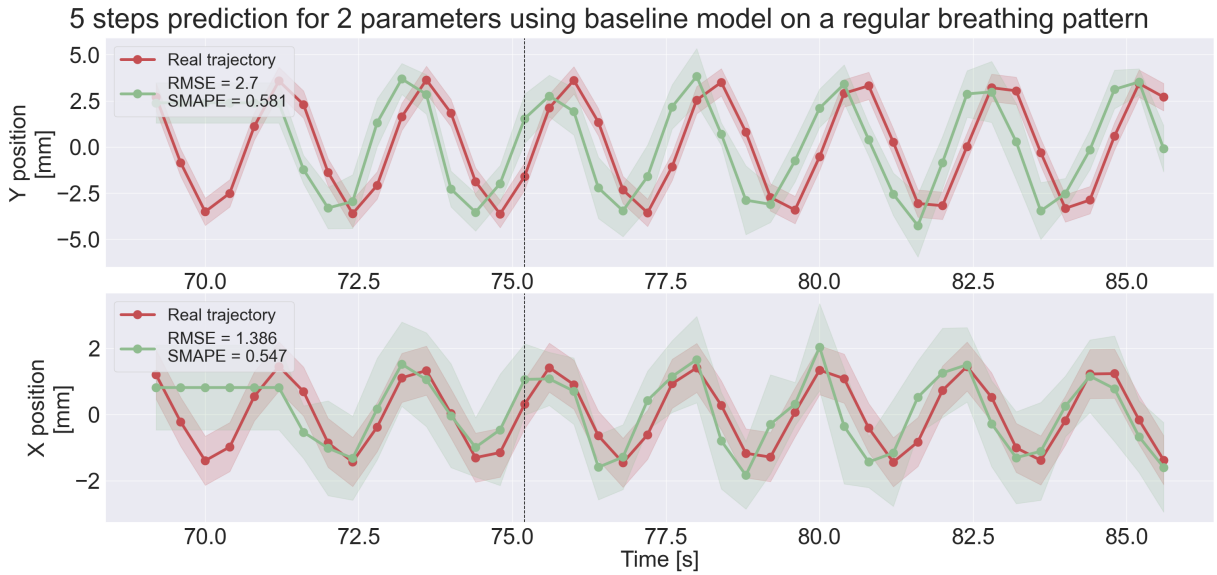

Figure 5.4: 5 steps prediction for 2 parameters using baseline model on a regular breathing pattern. The mean real trajectory is represented in red and the mean prediction in green, both with their variance represented in lighter color. The regular breathing is simulated by two sinus: $y_{simple} = 2\sin(4.57t), x_{simple} = -0.8\sin(4.8t)$. P1C. Testing set. 100 simulations.

2.1.2 Real trajectory

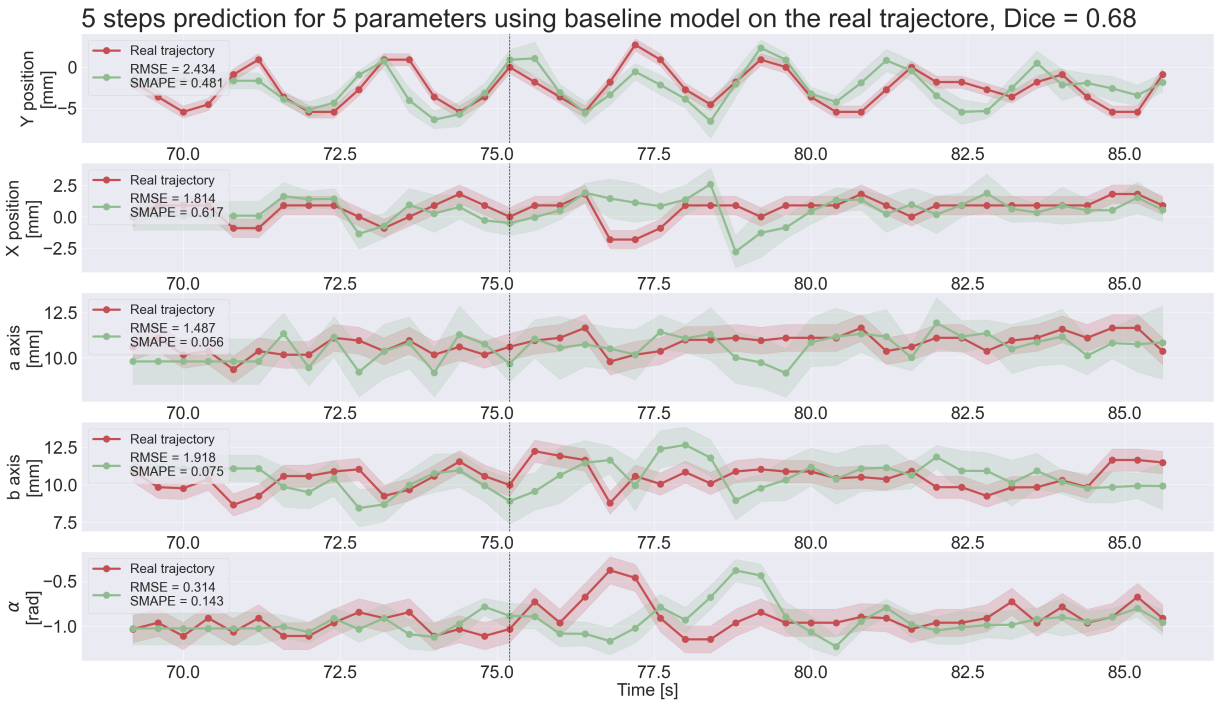

Figure 5.5: 5 steps prediction for 5 parameters using baseline model on the real trajectory. The mean real trajectory is represented in red and the mean prediction in green, both with their variance represented in lighter color. P1C. Testing set. 100 simulations.

For this application, a bigger contour is requested to be sure to irradiate the entire tumor. To be sure that the predicted ellipse is not too small, we try to dilate the predicted ellipse. The dilation is applied with a kernel of 3×3 and 5×5. It will increase the surface covered by the prediction from 1 and 2 pixels respectively to each side. The boxplot represented in Figure 5.6 enables to determine if the predicted ellipse needs to be dilated or not in order to be a better prediction thanks to the Dice coefficient.

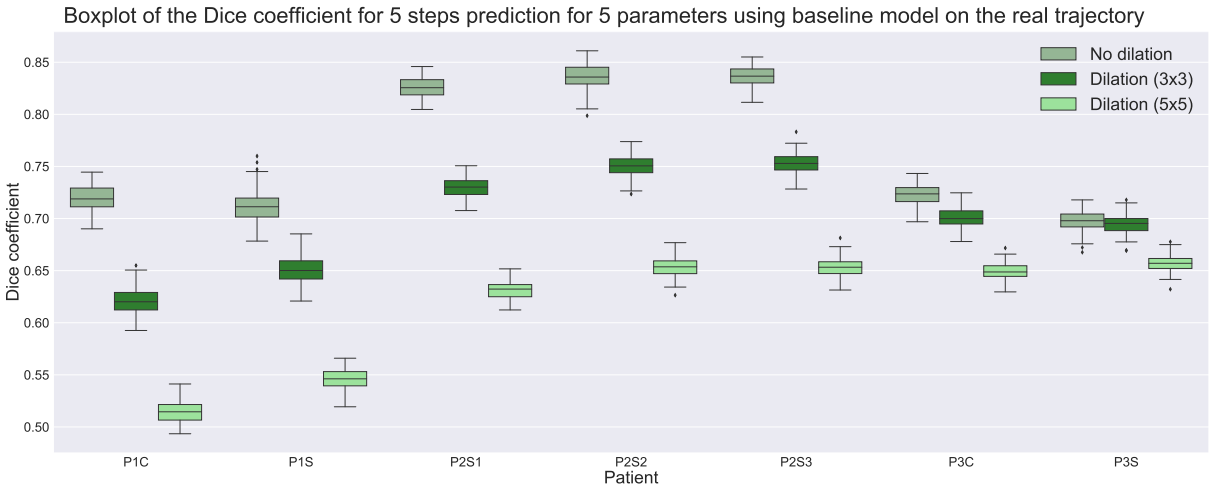

Figure 5.6: Boxplot of the Dice coefficient for 5 steps prediction for 5 parameters using baseline model on the real trajectory. The dilation kernel applied on the predicted ellipse varies between 1×1, 3×3 and 5×5. All patients. Testing set. 100 simulations.

The Figure 5.4 shows that the baseline prediction is simply the measurements translated by 5 time steps. The RMSE of the Y position is higher than the one of the X position while the SMAPE is around the same value (0.5).

In Figure 5.5, we can also see that the prediction is the translation of the measurements. The SMAPE of the Y position (0.48) is lower than the one of the baseline model (0.58) but the SMAPE of the X position is higher (0.62 vs 0.54). The SMAPE of the a and b axes as well as α are lower (0.05, 0.07, 0.1).

The Figure 5.6 shows that the Dice coefficient decreases when a dilation is applied. Without dilation, the Dice coefficient is above 0.7 for each patient.

The advantage of the baseline model is that the amplitude is in the same range as the real amplitude since we are dealing with measures translation. Only the noise can influence the amplitude of the predicted trajectories.
In addition, the accuracy of the prediction depends on the signal period and the prediction step. If the prediction time is a multiple of the signal period, the prediction will be very performant as we will be at the same moment inside the period in a next period as illustrated in Figure 5.7. If this happens, the magnitude of the error between the real trajectory and the prediction will be the noise added on the data. Since the relative error is quite similar for X and Y predictions, we can conclude over the dependence of the RMSE on the amplitude. Indeed as the Y amplitude is higher, the error will always be bigger than for the X position. The SMAPE is thus a good metric to compare different parameters that have not the same amplitude.

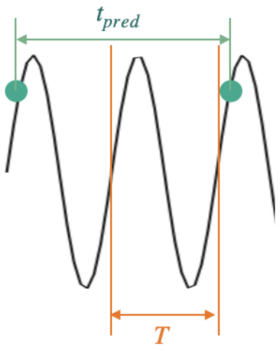

Figure 5.7: Illustration of the dependence between the prediction accuracy of the baseline model and the signal period.

The fact that the SMAPE of the X position is higher can be explained by the more sinusoidal character of the Y position. The lower error of α can be explained by the trajectory that contains less variation. For a and b axes, the reason can be that the variations in the trajectory arrive slower than for the X position, there is no huge peak. The error at each time step is thus smaller.

For all patients, the predicted ellipses do not need to be dilated as shown in Figure 5.6. This can be because the average amplitude corresponds to the amplitude of the real trajectory.

2.2 Sinusoidal model

As explained in the Section 2.1 of Chapter 2, two different models of prediction are used. The following figures 5.8 and 5.9 allow to conclude about the best sinusoidal model to predict regular and irregular trajectories. The trajectories of the evaluation set are compared qualitatively and quantitatively thanks to the RMSE, the SMAPE and the Dice coefficient.

2.2.1 Regular breathing pattern

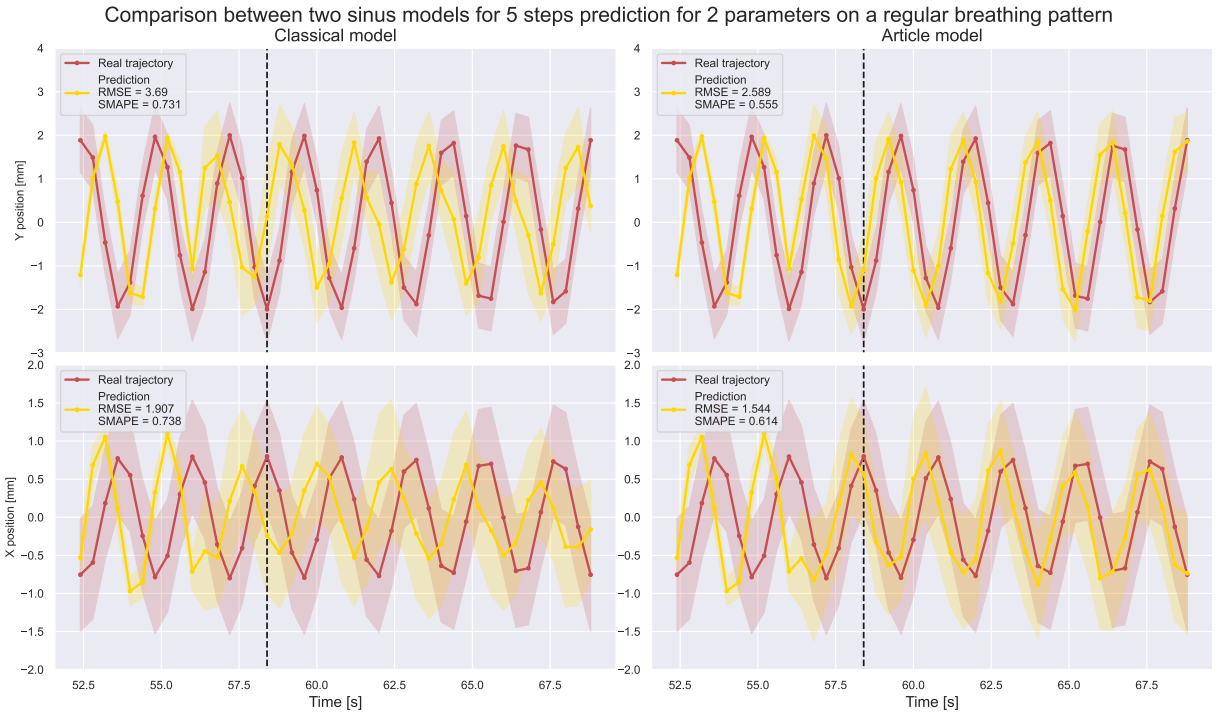

Figure 5.8: Comparison between two sinus models for 2 parameters on a regular breathing pattern. The mean real trajectory is represented in red and the mean prediction in yellow, both with their variance represented in lighter color. Left: Classical method. Right: Method that comes from the article [50]. The regular breathing is simulated by two sinus: $y_{simple} = 2\sin(4.57t), x_{simple} = -0.8\sin(4.8t)$. P1C. Evaluation set. 50 simulations.

2.2.2 Real trajectory

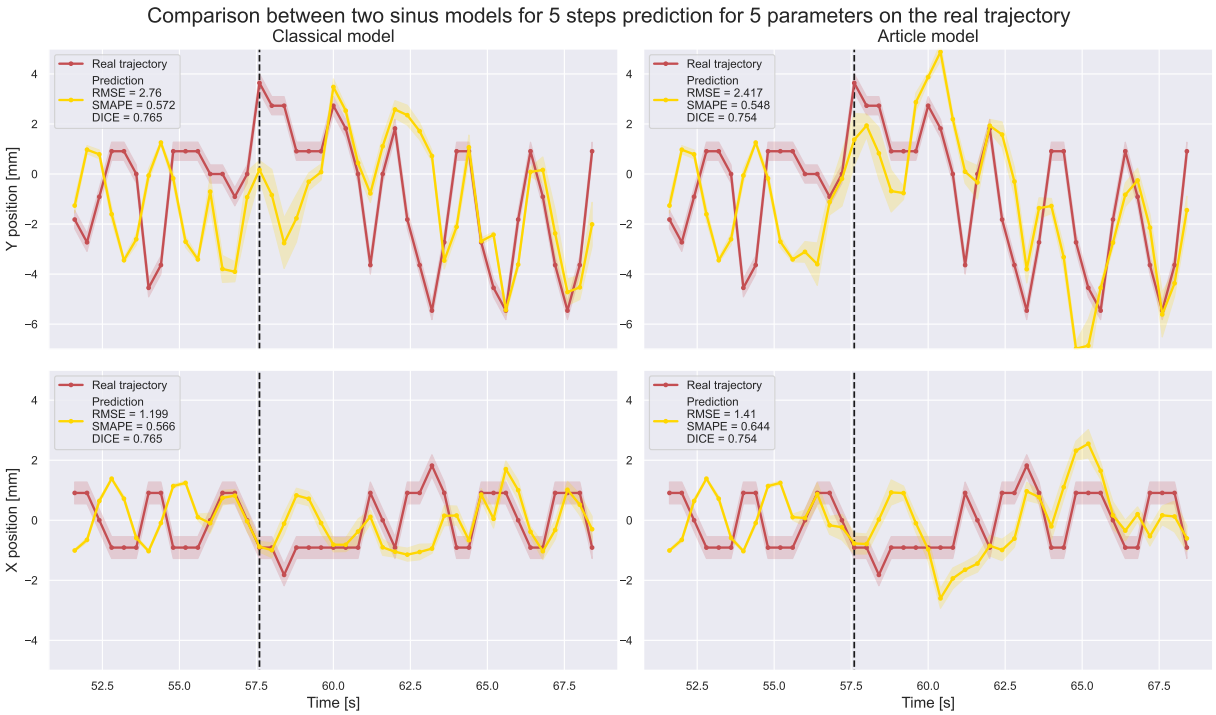

Figure 5.9: Comparison between two sinus models for 5 steps prediction for 5 parameters on the real trajectory. Left: Classical method. The mean real trajectory is represented in red and the mean prediction in yellow, both with their variance represented in lighter color. Right: Method that comes from the article [50]. P1C. Evaluation set. 50 simulations.

Moreover, the number of periods considered to compute the parameters of the sinusoidal signal (ω, ϕ, d and A) is varied to determine its influence on the prediction. The Figure 5.10 allows to determine the optimal number of periods based on the Dice coefficient between the physician contour and the predicted ellipse.

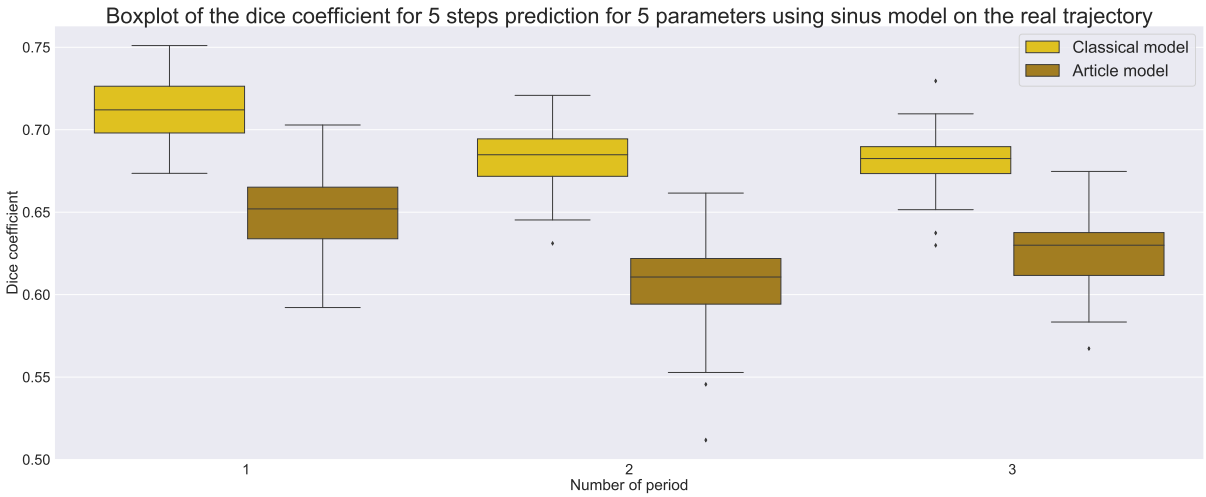

Figure 5.10: Boxplot of Dice coefficient for 5 steps prediction for 5 parameters using sinus classical model on the real trajectory. The number of periods used for the prediction varies from 1 to 3. P1C. Evaluation set. 50 simulations.

Once the best number of periods is chosen, the RMSE between the predicted trajectory and the real one can be dressed for both methods as well as the Dice coefficient to select the best sinus model for all patients. The results can be seen in a quantitative way in Figures 5.11 and B.1. The Table 5.2 summarizes the results for the optimal value of the number of periods, the optimal method between both presented and the corresponding RMSE and SMAPE for all other patients.

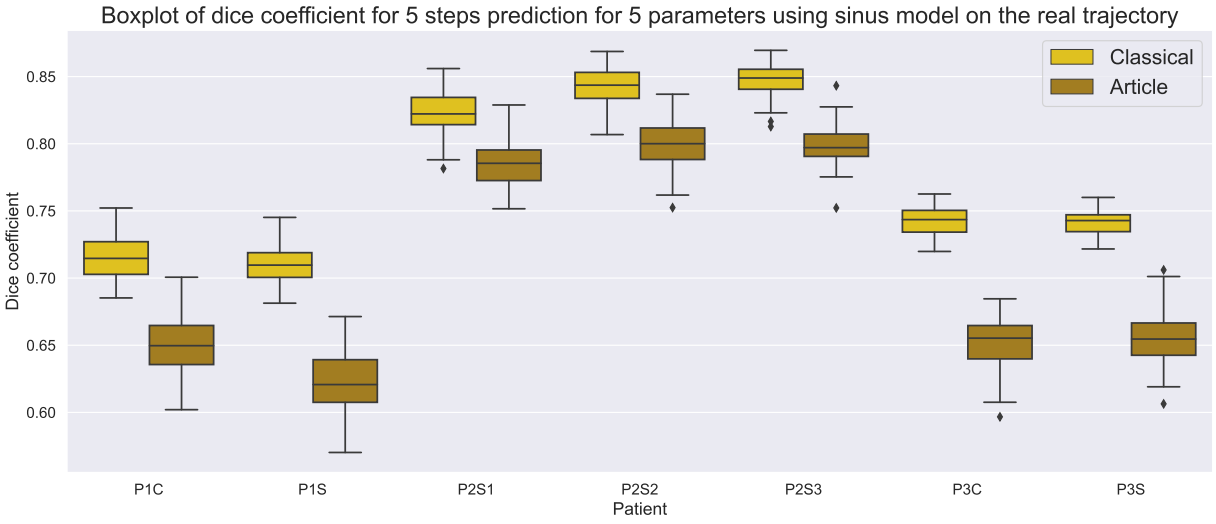

Figure 5.11: Boxplot of Dice coefficient for 5 steps prediction for 5 parameters using two sinus models on the real trajectory. All patients. Evaluation set. 50 simulations.

			Y position		X position	
Patient	Optimal method	Optimal $\#T$	RMSE [mm]	SMAPE	RMSE [mm]	SMAPE
P1C	Classical model	1	3.581	0.622	1.908	0.643
P1S	Classical model	1	3.381	0.547	2.007	0.494
P2S1	Classical model	1	2.777	0.406	1.384	0.562
P2S2	Classical model	1	2.791	0.485	1.302	0.784
P2S3	Classical model	1	2.886	0.328	1.642	0.770
P3C	Classical model	1	5.846	0.680	2.469	0.729
P3S	Classical model	1	5.142	0.627	2.683	0.566

Table 5.2: Table containing the optimal method and number of periods for each patient, and the corresponding RMSE and SMAPE. $\#T$ refers to the number of periods.

On a regular breathing pattern, the Figure 5.8 shows that the RMSE of both Y and X positions is lower for the article model. The SMAPE is also smaller, especially for the Y position.

We can observe in Figure 5.9 that the predicted trajectory of the Y position with the classical model is better at the end, between 63.75 s and the end of the trajectory.
In general, when an irregularity occurs, it is not directly taken into account as shown by the fact that it occurs later in the prediction. As an irregularity occurs, the prediction sometimes decreases its amplitude, it thus reduces the error. Indeed, in the article model,

it takes into account the noise and thus continues large predictions.

The Figure 5.10 shows that taking one period to determine the parameters of the sinus model gives the best results.

In Figures B.1 and 5.11, we can see that the RMSE is higher for the article method and the Dice coefficient is lower. For the other patients, we can see that the mean error is lower than 4 and 3 mm (Y and X respectively) except for P3 for which it is quite big (cf. Table 5.2). The RMSE of Y position for both coronal and sagittal views are large: 5.1 and 5.8 mm; and the SMAPE are: 0.68 and 0.63. The errors on the X position are also large.

On a regular sinusoidal trajectory, the article model gives better performance. It is quite surprising that the classical model does not work so well knowing that the trajectory is a perfect sinus. This can be due to the noise contained in the data. The classical model does not succeed to find correctly the parameters. The article model has an idea about the noise value because it takes into account the difference between the prediction and the real value at the present time.

On a real trajectory, the prediction is better when using the classical model. Again, this observation is unexpected. Because the trajectory is not a perfect sinus, it was expected that the article model would be better. The article model has the advantage of taking into account the error of the prediction at the current time. When there are less irregularities, the classical model gives lower error in the prediction which seems logical due to the sinusoidal character of the trajectory at the end of the set.

Taking one period to determine the parameters of the model seems to give the best results. The reason could be that the model can catch the irregularities occuring over time. But the problem by taking only one period is that the consequence will be felt one period after. On the contrary, if too many periods are taken, a mean is done and the irregularities are thus smoothed. In addition, the classical model remains the best one. This is the reason why we decide to use the classical model for the rest of the prediction with a sinusoidal model.

The big error for P3 especially is due to the fact that the trajectory of this particular patient is at one time very irregular (cf. Figure A.5) and depending on how the data sets are cut, it can lead to erroneous frequency calculations.

Figure 5.12 allows to provide a visualisation of the performance of the optimal sinusoidal model on the P1C testing set. The quantitative character of the performance and the accuracy of the optimal model found for the other patients testing set will be observed thanks to the Dice coefficient in the algorithms comparison.

As for the baseline model, the predicted ellipse is dilated to see whether it covers better the real contour of the tumor. This is the purpose of the Figure 5.13.

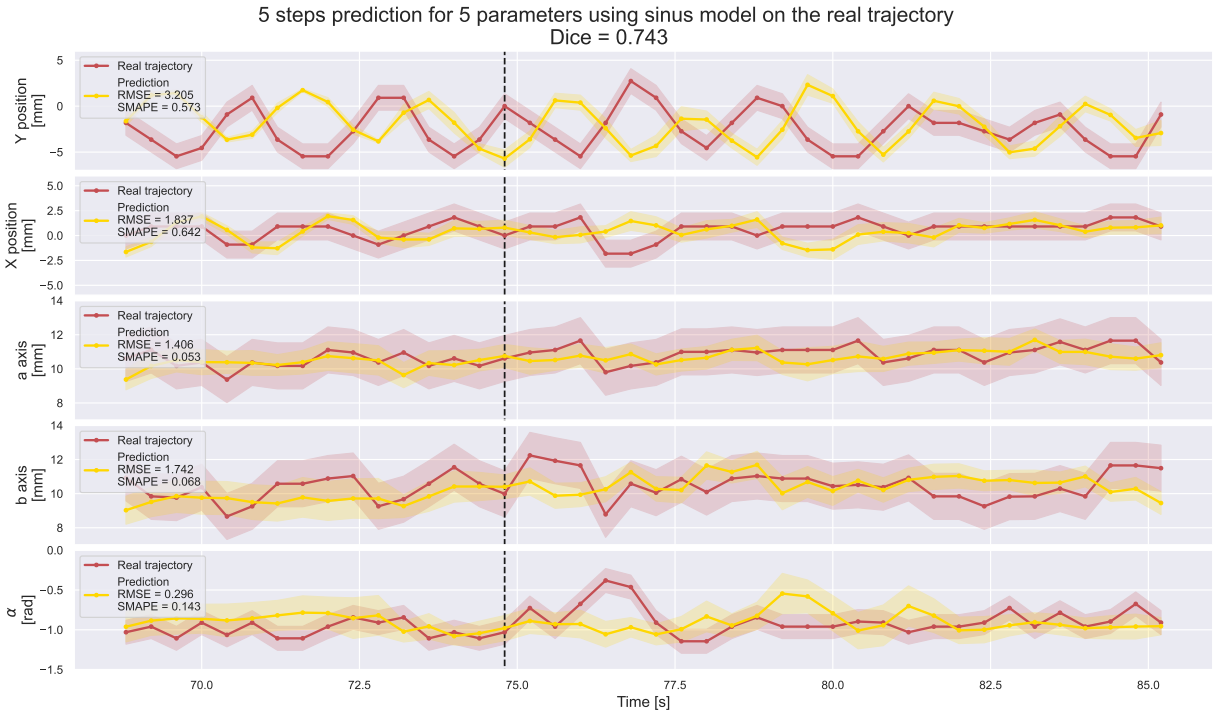

Figure 5.12: 5 steps prediction for 5 parameters using classical sinus model on the real trajectory. The mean real trajectory is represented in red and the mean prediction in yellow, both with their variance represented in lighter color. P1C. Testing set. 100 simulations.

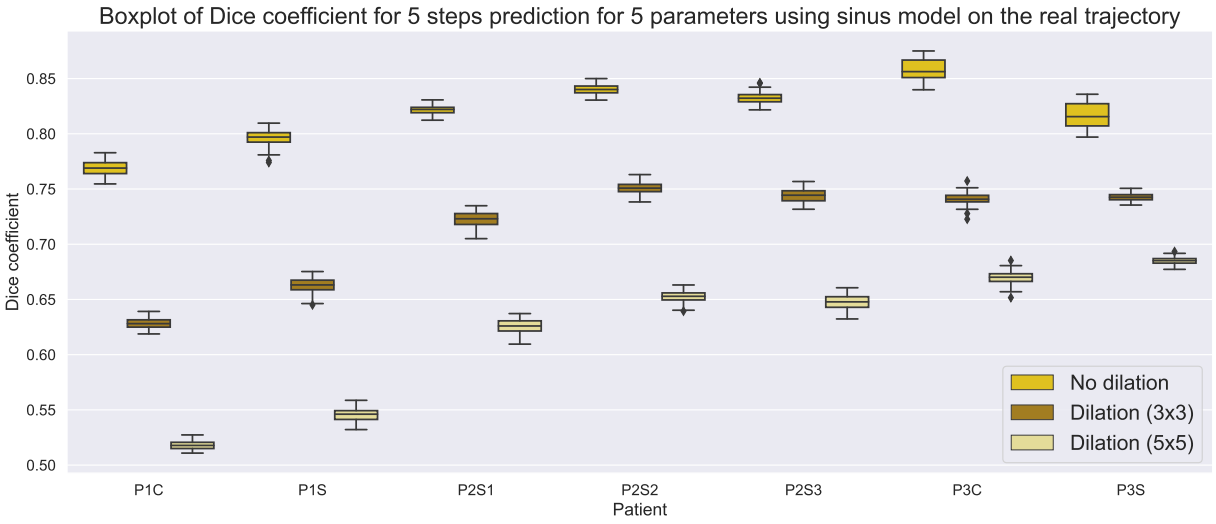

Figure 5.13: Boxplot of Dice coefficient for 5 steps prediction for 5 parameters using classical sinus model on the real trajectory. The dilation kernel applied on the predicted ellipse varies between 1×1, 3×3 and 5×5. All patients. Testing set. 100 simulations.

In Figure 5.12, the SMAPE for Y and X positions are a little bit higher than the ones on the training set. As with the baseline model, the errors on a and b axes as well as on α are smaller and this is confirmed with the SMAPE values.

Concerning the prediction, we can observe that the range of predicted amplitude is the same as the one of the real trajectory. The periods are also well approximated. However, the irregularities are not taken into account directly. Moreover, the Y trajectory seems out of phase when a more regular pattern occurs.

The predicted ellipse is better if no dilation is applied as we can see in Figure 5.13. The Dice coefficient of the predicted ellipse is above 0.75 without dilation for all patients, which is higher than the one obtained with the baseline model.

In Figure 5.12, the Y predicted trajectory seems out of phase when a more regular pattern occurs. This leads to the conclusion that the dephasage is certainly not correctly considered. It may be due to the fact that we reconstruct a sinus when predicting (this sinus start at $t = 0$ s) while the time is continuously growing as we evolve in the trajectory.

Based on the Dice coefficient, we can conclude that the sinusoidal model has better performance than the baseline model.

2.3 Kalman filter

To present the Kalman predictor, we need a model that represents the dynamics of the tumor movement. To obtain this model, two strategies are tested: an analytical computation and a data-based method by using the system identification toolbox.

2.3.1 Analytical

In the first strategy outlined, one parameter c can be optimized to represent as well the dynamics. The trajectories are represented on the evaluation set for different values of this parameter c and the Dice coefficients between the real contour and the predicted one obtained with the Kalman filter are represented in a boxplot. The Figures 5.14 and 5.15 as well as Figures 5.17 and 5.18 allows to decide the optimal parameter c for a regular breathing pattern and a real trajectory respectively.

Center position: (Y), regular breathing pattern

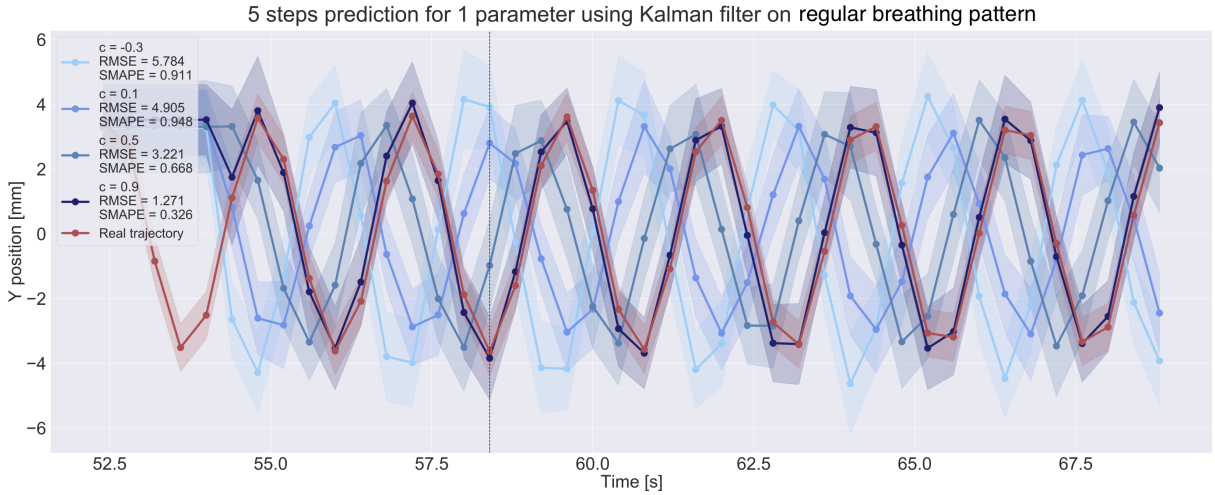

Figure 5.14: 5 steps prediction for 1 parameter using Kalman on regular breathing pattern. The mean real trajectory in red, the measurements in light red and the prediction in blue. The model is found analytically. Four values of c are used: $c = -0.3, 0.1, 0.5$ and $c = 0.9$. P1C. Evaluation set. 50 simulations.

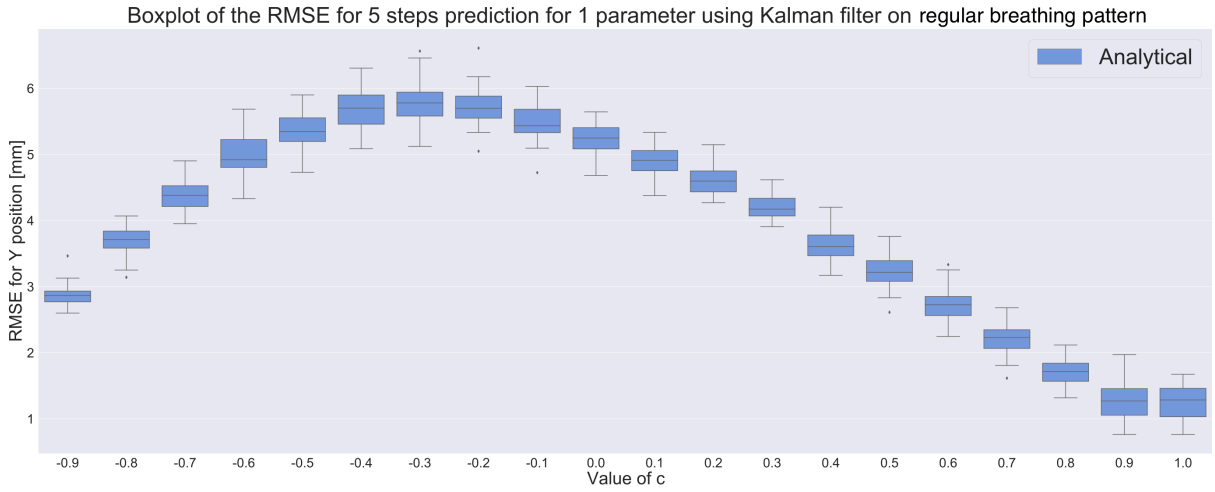

Figure 5.15: Boxplot of the RMSE for 5 steps prediction for 1 parameter using Kalman filter on regular breathing pattern. The model is found analytically. The parameter c varies between -0.9 and 1.0. P1C. Evaluation set. 50 simulations.

The Figure 5.16 shows the prediction on the testing set for a regular breathing pattern. This allows to evaluate the performance of the system found and determine the presence of overfitting.

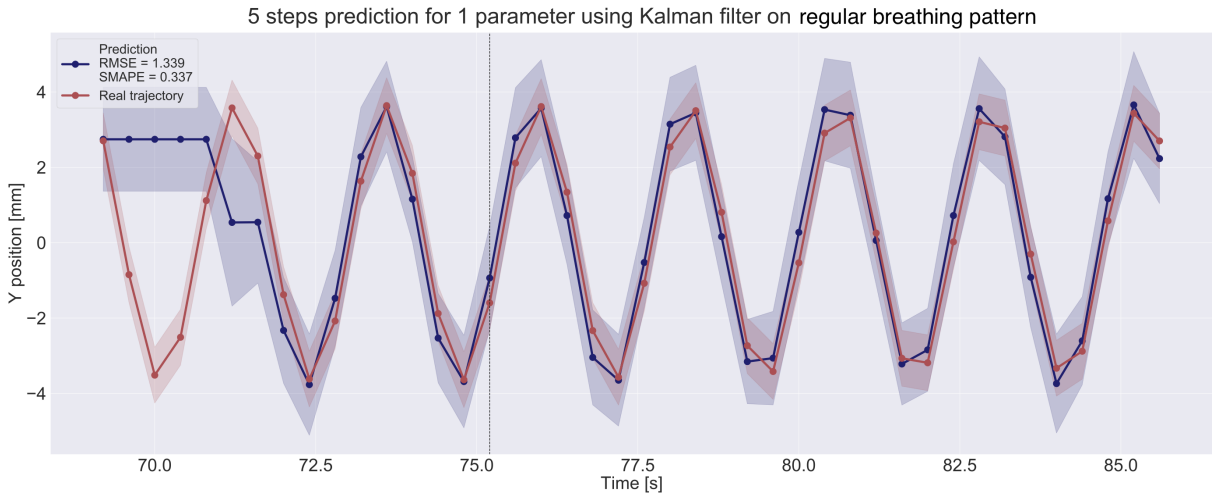

Figure 5.16: 5 steps prediction for 1 parameter using Kalman filter on regular breathing pattern. The mean real trajectory in red, the measurements in light red and the prediction in blue. The model is found analytically. P1C. Testing set. 100 simulations.

In Figure 5.15, a minimum can be observed when $c = 0.9$. This is verified in Figure 5.14, we see that the dark blue curve follows the real trajectory represented in red. Moreover, we observe that when the parameter c varies, the predicted trajectory is shifted horizontally but its frequency does not change. This can also be observed in Figure C.1 for P3C.

In Figure 5.16, we can see that until $t = 71$ s, the prediction is just a constant. Between $t = 71$ s and $t = 72.5$ s, the prediction does not follow perfectly the real trajectory. After $t = 72.5$ s, the predicted trajectory follows the real one.
The RMSE and the SMAPE with this method on a regular trajectory are smaller than the ones with the baseline model (RMSE = 1.34 vs 2.7 mm, SMAPE = 0.33 vs 0.6) and the sinus model (RMSE = 2.6, SMAPE = 0.55). Moreover, the performance on the training (RMSE = 1.27 mm) and testing (RMSE = 1.34) sets are similar. We can see the same tendency for the SMAPE.

This shows that, contrary to our first guest, the parameter c is not related to the frequency of the signal but to the phase shift.

What we can conclude is that this parameter is quite easy to optimise since a minimum value is clearly visible.

This shows that, contrary to our first guest, the parameter c is not related to the frequency of the signal but to the phase shift. However, more mathematically, when we solve the system of differential equations 2.17, the solutions are quite complex. The change of c value induces change in more than one parameter in the solution. It is thus complex to conclude it visually.

What we can conclude is that this parameter is quite easy to optimise since a minimum value is clearly visible.

The first predictions are not really predictions because the Kalman predictor need data to work. This represents the time between the start of the prediction and the time at which the first prediction occurs. After that, the prediction does not fit perfectly the trajectory because the filter is gaining information. Indeed, it is written in the litterature that the Kalman filter needs time to acquire information. This time is called the adaptation time. Because it is assumed to last 15 measures, the RMSE and SMAPE are calculated from the 16th measure. This adaptation time will be more understood in the following results.

Because the errors are the same on the training and testing sets, we can conclude that there is no overfitting.

Center position: (Y), real trajectory

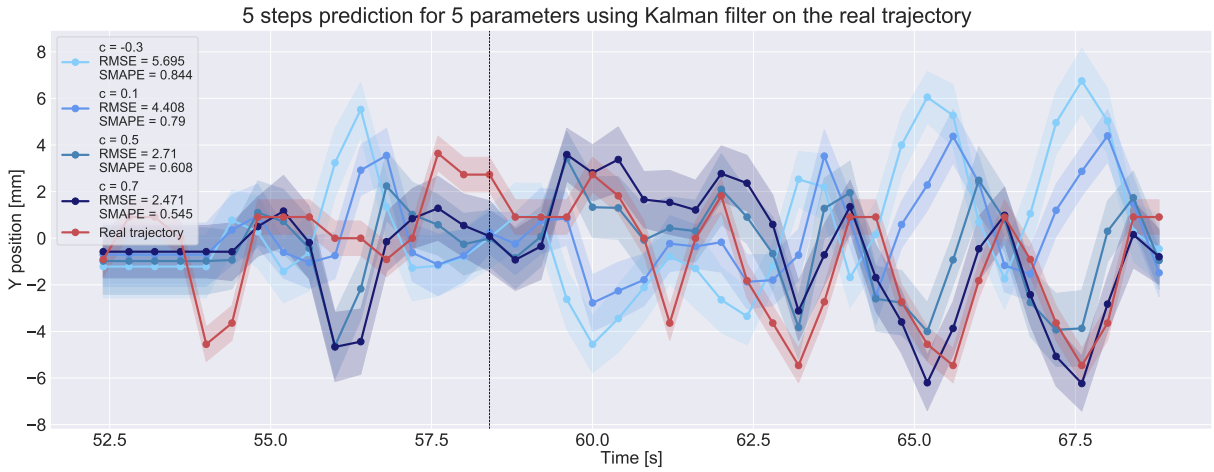

Figure 5.17: 5 steps prediction for 1 parameter using Kalman filter on the real trajectory. The mean real trajectory in red, the measurements in light red and the prediction in blue. The model is found analytically. Four values of the parameter c are used: $c = -0.3$, $c = 0.1$, $c = 0.5$ and $c = 0.7$. P1C. Evaluation set. 50 simulations.

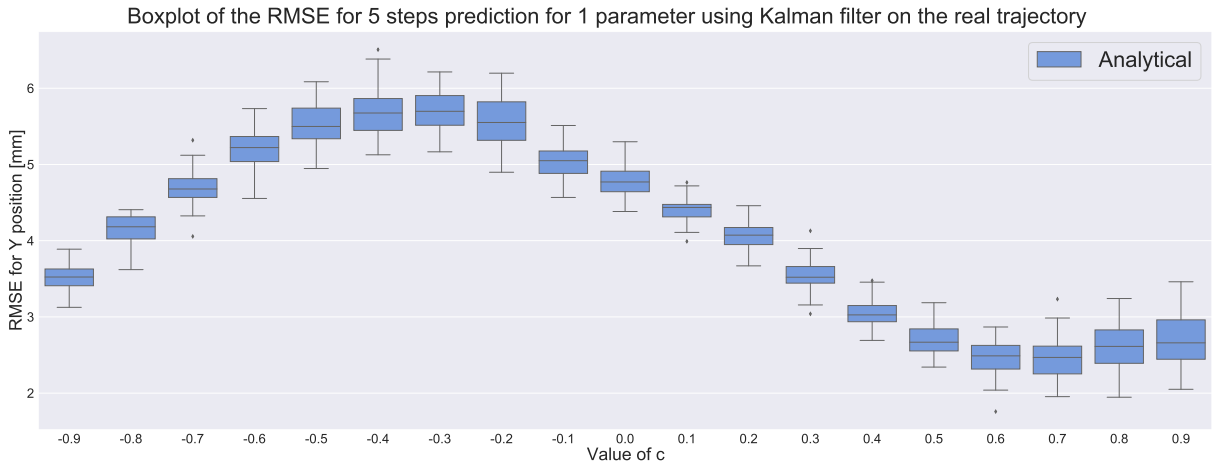

Figure 5.18: Boxplot of the RMSE for 5 steps prediction for 1 parameter using Kalman filter on the real trajectory. The model is found analytically. The parameter c varies between -0.9 and 1.0. P1C. Training set. 100 simulations.

The optimal c value and the corresponding RMSE and SMAPE are depicted for all patients in Table 5.3.

Patient	Optimal c	RMSE [mm]	SMAPE
P1C	0.7	2.447	0.555
P1S	0.7	2.265	0.585
P2S1	0.9	1.972	0.747
P2S2	-0.9	1.442	0.812
P2S3	0.9	2.146	0.475
P3C	0.8	2.517	0.431
P3S	0.8	2.759	0.450

Table 5.3: Table containing the optimal parameter c for each patient, and the corresponding RMSE and SMAPE.

The system with the optimal c value can be evaluated on the testing set for P1C in Figure 5.19. This allow to determine if there is overfitting.

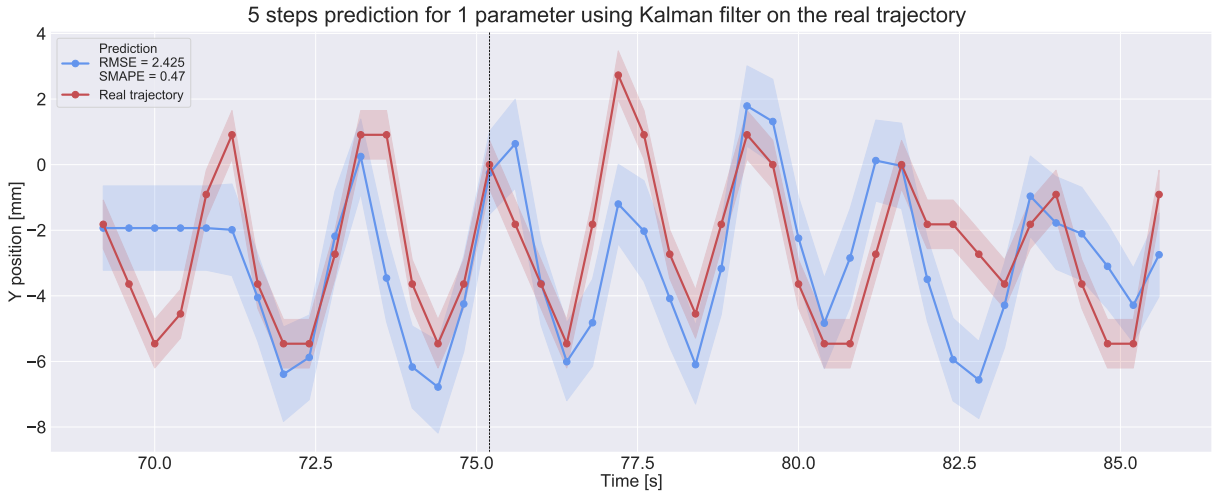

Figure 5.19: 5 steps prediction for 1 parameter using Kalman filter on the real trajectory. The mean real trajectory in red, the measurements in light red and the prediction in blue. The model is found analytically. Optimal parameters are used. P1C. Testing test. 100 simulations.

In Figure 5.17, we can see that there is a minimum of the Dice coefficient when $c = 0.7$. The dark blue line in Figure 5.17 is the associated prediction. At the start of the trajectory (around 55 s), the irregularities are predicted just a bit later than the time they really occur.

Moreover, even though it is not visible at first sight, we can observe a link between c and a horizontal shift. Indeed, the dark blue curve ($c = 0.7$) and the curve $c = 0.5$ seem shifted. The curve $c = 0.5$ seems also shifted with a slight variation of shape compared to the real trajectory. The curves $c = -0.3$ and $c = 0.1$ are horizontally shifted and seem to have undergone a symmetry of axis x compared to the real trajectory. In a sinus, this corresponds to a horizontal shift of a half period.

An adaptation time is also observed on the regular trajectory. Indeed, the prediction is better at the end of the trajectory.

In Table 5.3, we observe that the optimal c is high, it varies between 0.7 and 0.9 for all patients except for P2S2 for which it is equal to -0.9. The SMAPE increases to 0.8 for P2S2 and the smallest one is around 0.4.

In Figure 5.19, the RMSE and the SMAPE are higher than the ones on a regular breathing pattern.

However, the performances of Kalman filter are the same compared to the baseline model on the real trajectory. The RMSE and SMAPE of Kalman filter equal respectively 2.44 mm and 0.47 mm while they equal 2.4 mm and 0.47 for the baseline model. The sinus model gives higher errors: RMSE = 3.2 mm and SMAPE = 0.57.

We can see that the prediction is quite good in general but the quality of the prediction decreases when a high irregularity occurs as it is the case around $t = 82.5$ s.

The trend is the same as the one observed for a regular breathing pattern since the higher values of c gives better results. Based on the previous observations, we can thus conclude that the parameter c is related to a horizontal shift.
The best prediction at the end can also be explained by the shape of the trajectory. We can see that it has a more sinusoidal shape and is thus more regular.

The same performances of the Kalman filter and the baseline model can be explained by the trajectory period (2.4 s) that more or less equals the prediction time step (2 s).
Again, we can say that there is no overfitting. Indeed, the RMSE on the testing and training sets are similar: 2.47 and 2.43 mm. The same trend can be observed for the SMAPE.

Ellipse parameters: (Y, X, a, b, α), real trajectory We decide to try to predict each parameter independently of the others using the analytical system in the Kalman predictor. The results are shown in Figure 5.20.

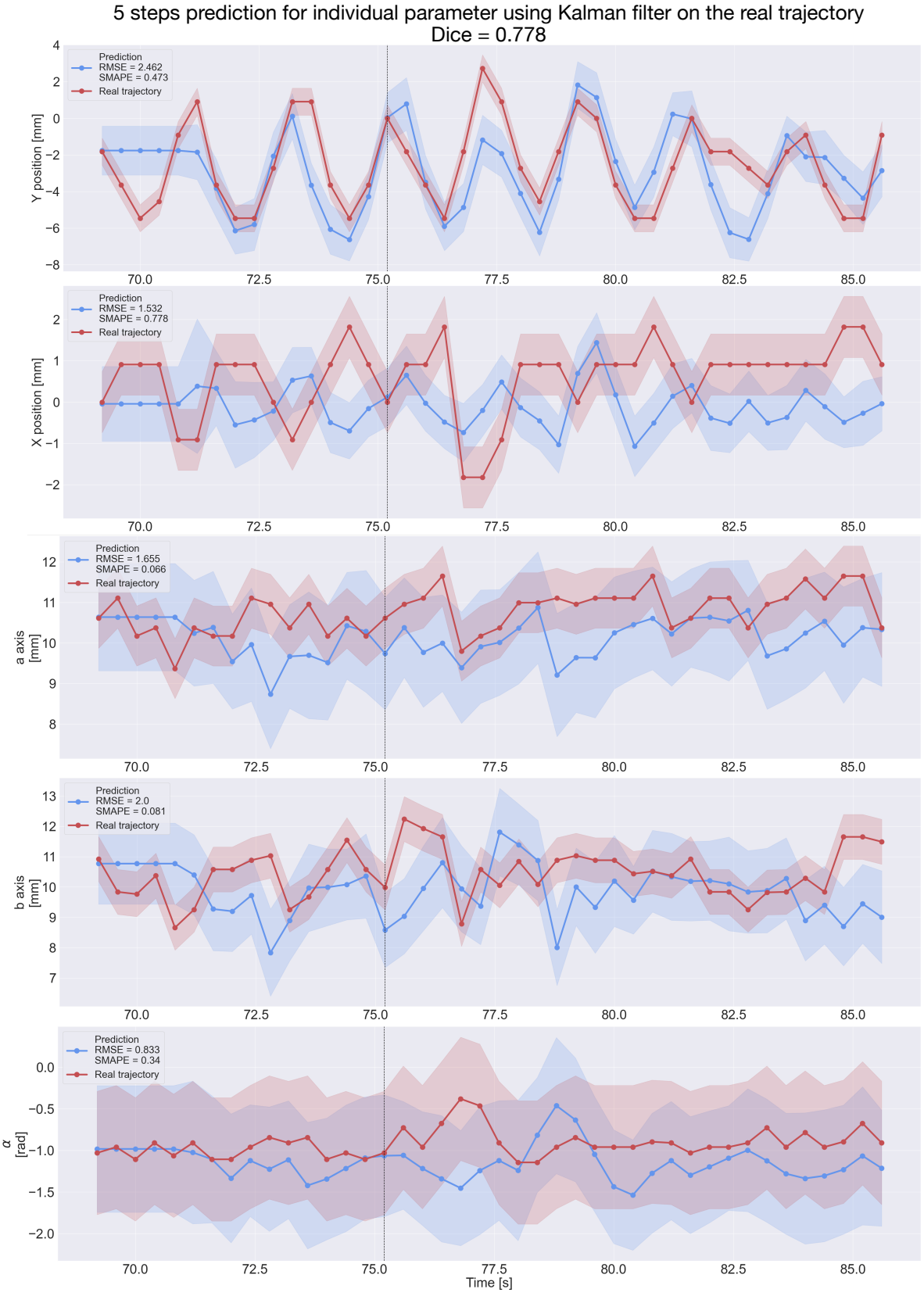

Figure 5.20: 5 steps prediction for 5 parameters using Kalman filter on the real trajectory. The mean real trajectory in red, the measurements in light red and the prediction in blue. The model of each parameter is independent of the others and is found analytically. Optimal parameters are used. P1C. Testing test. 100 simulations.

In Figure 5.20, we can see that the Y position is predicted correctly while the X position is not as accurate. Compared to other algorithms, we have RMSE of 2.462, 1.532, 1.655, 2.0 and 0.833 mm. In terms of center position and b axis, it has the best result through the methods that we have already seen while it has the worst one for the angle α.
The predicted trajectories for a, b and α seem shifted compared to the real ones while this observation can not be made for the X position.

The worst prediction of the X position can be due to the trajectory that is less regular. While for the α, the variation is not pronounced and a constant prediction would create less errors. However here the Kalman filter try to catch all small variations and is thus not very constant.

The shifts observed indicate that the Kalman filter takes into account the measurements.

2.3.2 System identification

The second technique consists of using a system identification toolbox and the data to find the general structure of the system matrices. This approach is first only used to predict the movement of the tumor center of mass (Y, X), and then the 5 parameters (Y, X, a, b, α).

With this approach, the order of the system is the parameter that is optimized. We evaluate the performance of the Kalman filter with systems of order 2 to 10. The Figure 5.21 allows to determine the optimal order for a regular breathing pattern in a visual and in a quantitative way thanks to the RMSE and the SMAPE. The RMSE for the Y and X positions of the predictions can be visualized in Figure D.1. The best order found is then used to generate a system and is evaluated on a testing set in Figure 5.22. This allows to determine if overfitting occurs. The same reasoning is hold for the two following parts (prediction of tumor center of mass and the 5 parameters on a real trajectory).

Center position: (Y, X), regular breathing pattern The performance of the Kalman filter using the system identification is first evaluated for a regular breathing pattern. This enables us to study the performance of the system identification for a periodic signal.

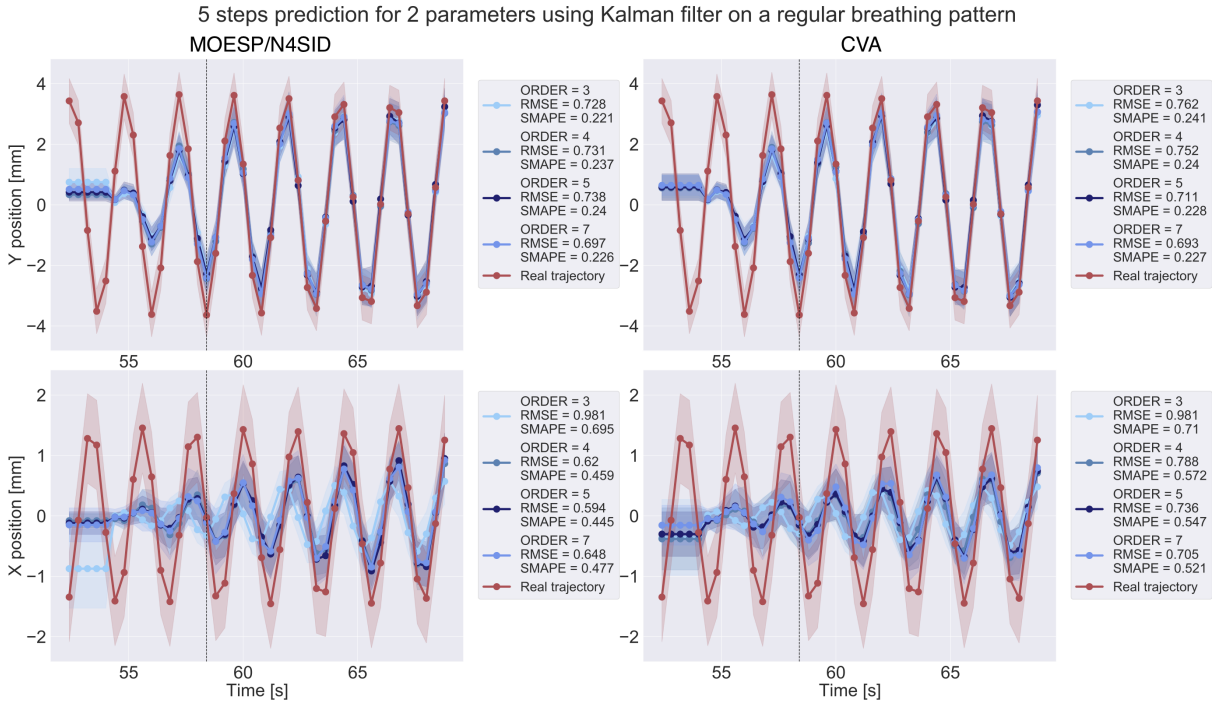

Figure 5.21: 5 steps prediction for 2 parameters using Kalman filter on a regular breathing pattern. The mean real trajectory in red, the measurements in light red and the prediction in blue. The model is found with a system identification and three identification methods are used. Left: MOESP/N4SID, Right: CVA. P1C. Evaluation set. 50 simulations.

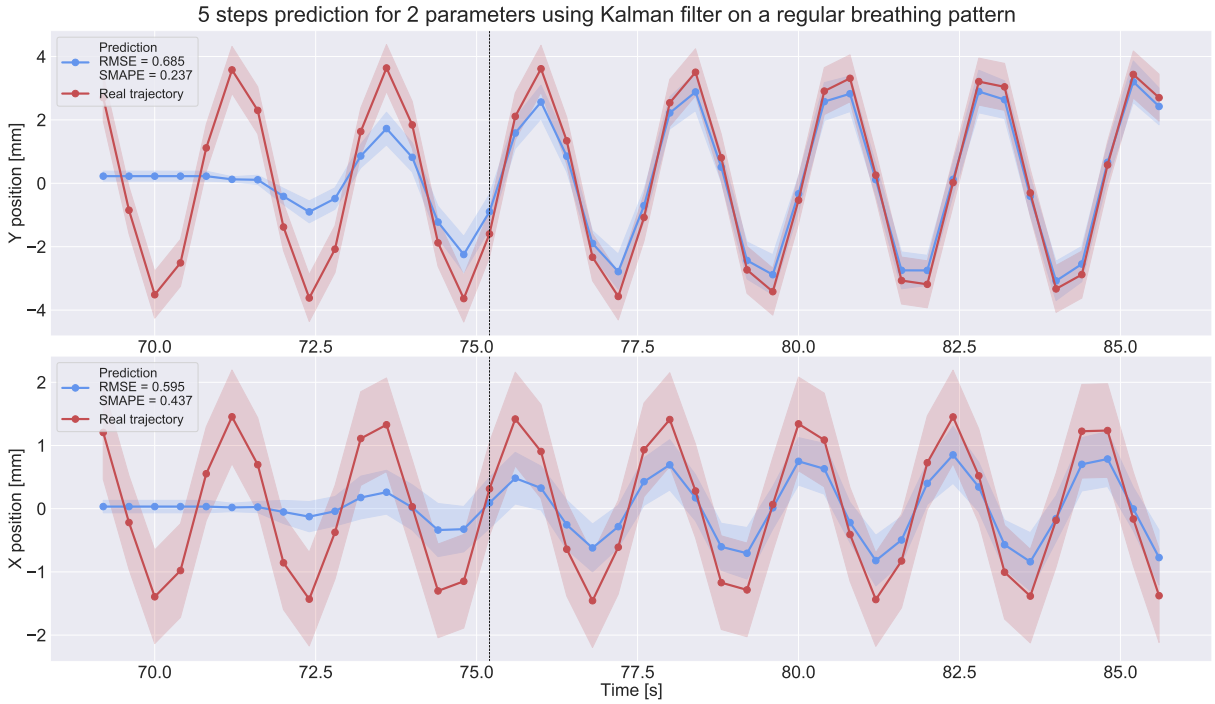

Figure 5.22: 5 steps prediction for 2 parameters using Kalman filter on a regular breathing pattern. The mean real trajectory in red, the measurements in light red and the prediction in blue. The model is found with a system identification. Optimal parameters are used. P1C. Testing set. 100 simulations.

Center position: (Y, X), real trajectory The results of predicting the tumor center of mass on a real trajectory are shown in Figures 5.23 and 5.24.

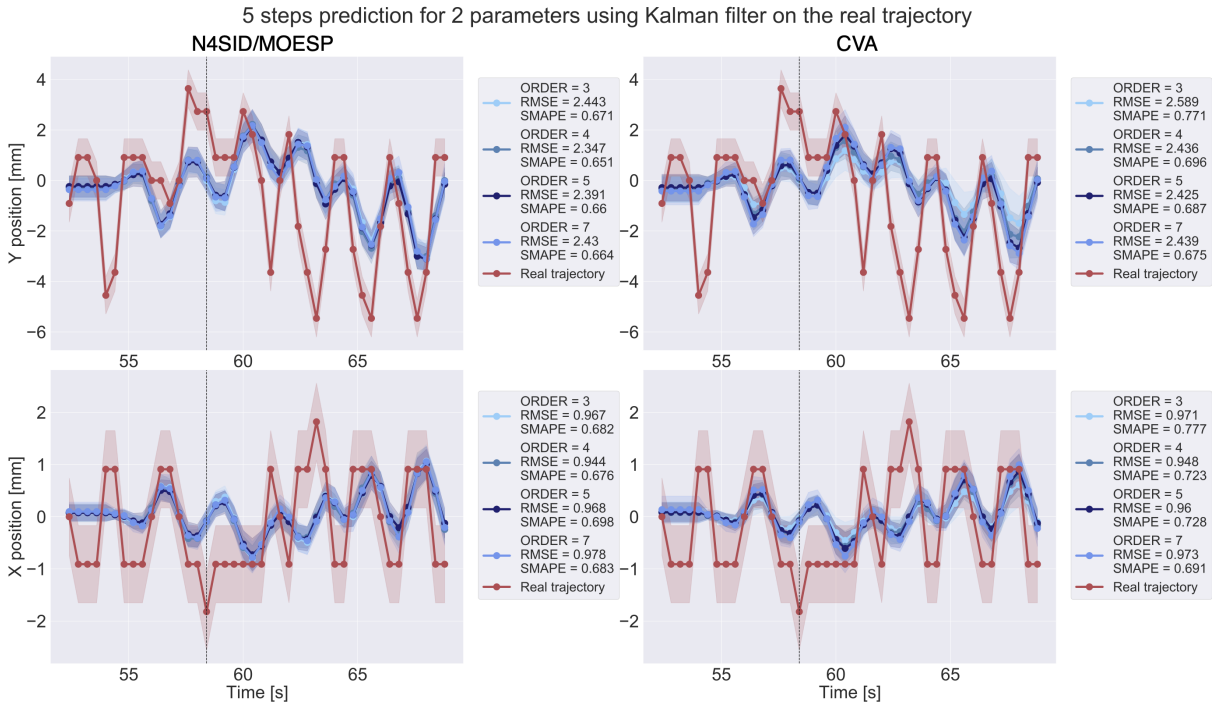

Figure 5.23: 5 steps prediction for 2 parameters using Kalman filter on the real trajectory. The mean real trajectory in red, the measurements in light red and the prediction in blue. The model is found with a system identification and three identification methods are used. Left: MOESP/N4SID, Right: CVA. P1C. Evaluation set. 50 simulations.

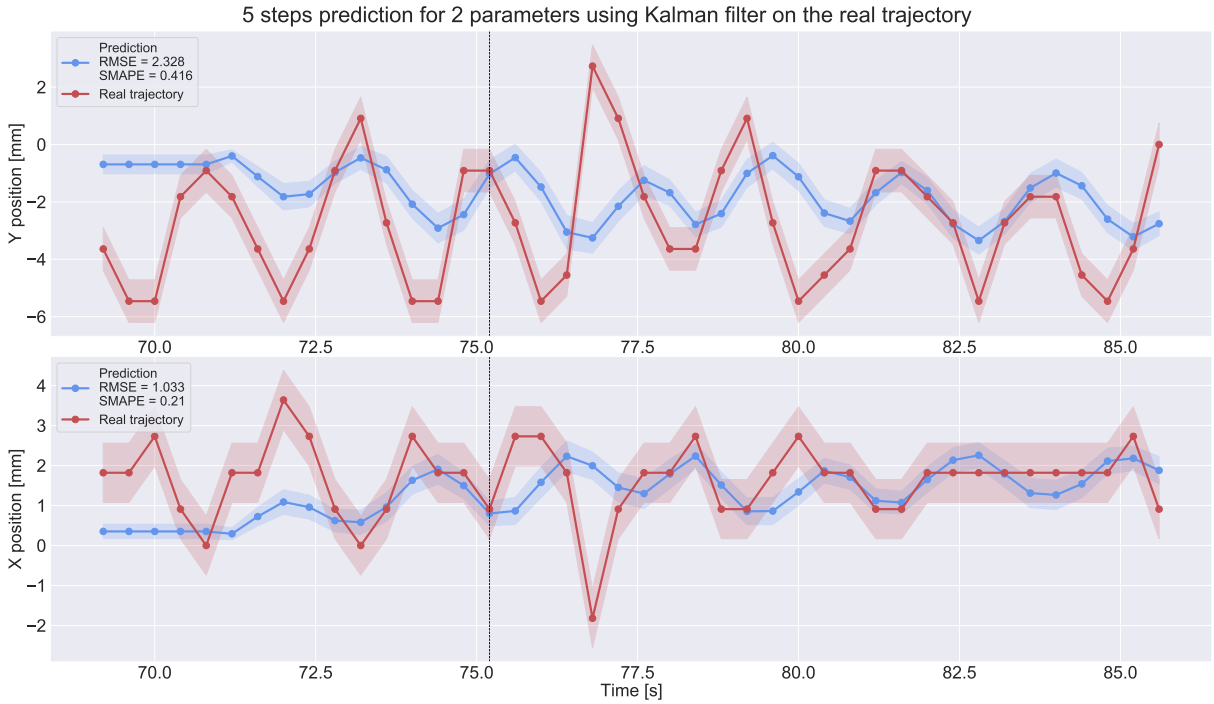

Figure 5.24: 5 steps prediction for 2 parameters using Kalman filter on the real trajectory. The mean real trajectory in red, the measurements in light red and the prediction in blue. The model is found with a system identification. Optimal parameters are used. P1C. Testing set. 100 simulations.

As shown in Figure 5.21, the MOESP and the N4SID identification methods give the same results. Indeed, we can see in Figure D.1 that the mean and the standard deviation of the RMSE for both positions are similar. The outcome of CVA is nearly the same as the one of both other methods, the RMSE are slightly higher.

There are only small differences between the RMSE and the predicted trajectories between all orders. In Figure D.1, we only observe bigger values of RMSE for order 2 and 3 for X position. From the order 4, there is no visible variation. In Figure 5.21, a certain time at the beginning of the set is highlighted during which the errors are bigger and the prediction is first very small and increases to reach the amplitude of the real signal. For the X position, this period seems longer as the prediction continues to improve at the end of the trajectory.

On the testing test shown in Figure 5.22, the prediction is quite good. The SMAPE for respectively the Y and the X positions are in the same range as the one of the training set. Indeed, the errors are smaller on the testing set (RMSE Y = 0.68 mm, RMSE X = 0.59 mm) than in the training set (RMSE Y = 0.73 mm, RMSE X = 0.62 mm). Again, there is an adaptation time which is quite longer for the X position.

The observations are now stated for the real trajectories case. We can observe in Figure 5.23 that N4SID and MOESP give the same results and CVA nearly the same. When comparing the different orders, we do not observe any trend. The RMSE and the SMAPE only vary a little but the order 4 is the best one due to a compromise between a small error and a small order to avoid overfitting.

We see that the prediction on the real trajectory in Figure 5.24 is less precise than the one of regular breathing pattern even after the 15 first measures. The prediction seems to follow a periodic trajectory and does not take into account a lot of irregularities appearing in the real trajectory. However, the RMSE and the SMAPE are in the same range as the one in the evaluation set.

The highlighted time at the beginning of the set corresponds to an adaptation time for the Kalman filter. Indeed, this algorithm needs a certain time to acquire information about the real evolution of the system and its dynamics. As a consequence, we calculate the errors from the 15th measures so that at this time, the Kalman filter has acquired enough information to predict a good trajectory. For the Y position, this period lasts 15 measures while for X it is a bit more but we are limited in the number of data here.

It could be interesting to train during a longer period of time to determine whether the prediction continues to evolve. Because the adaptation time is longer for the X position, the prediction is less performant and the errors are higher than for the Y position.

The order of the system does not influence a lot the prediction. This is due to the limited number of data we have since here, an higher degree means more chance to have overfitting. Indeed, here, we want an order not so big to avoid overfitting but not too low to be fairly accurate. It would be possible that the prediction becomes better if the testing set was composed of more measures since we could increase a bit the system order without unstability.

We can conclude that there is no overfitting with our model on a real trajectory as the performances on both sests are in the same order (SMAPE Y: 2.35 vs 2.33 mm, SMAPE X: 0.94 vs 1.03)

The prediction of the tumor center position on a real trajectory is not accurate but follows the variation in the trajectory.

Ellipse parameters: (Y, X, a, b, α), real trajectory The best order is determined visually and quantitatively by using Figures 5.25 and 5.26.

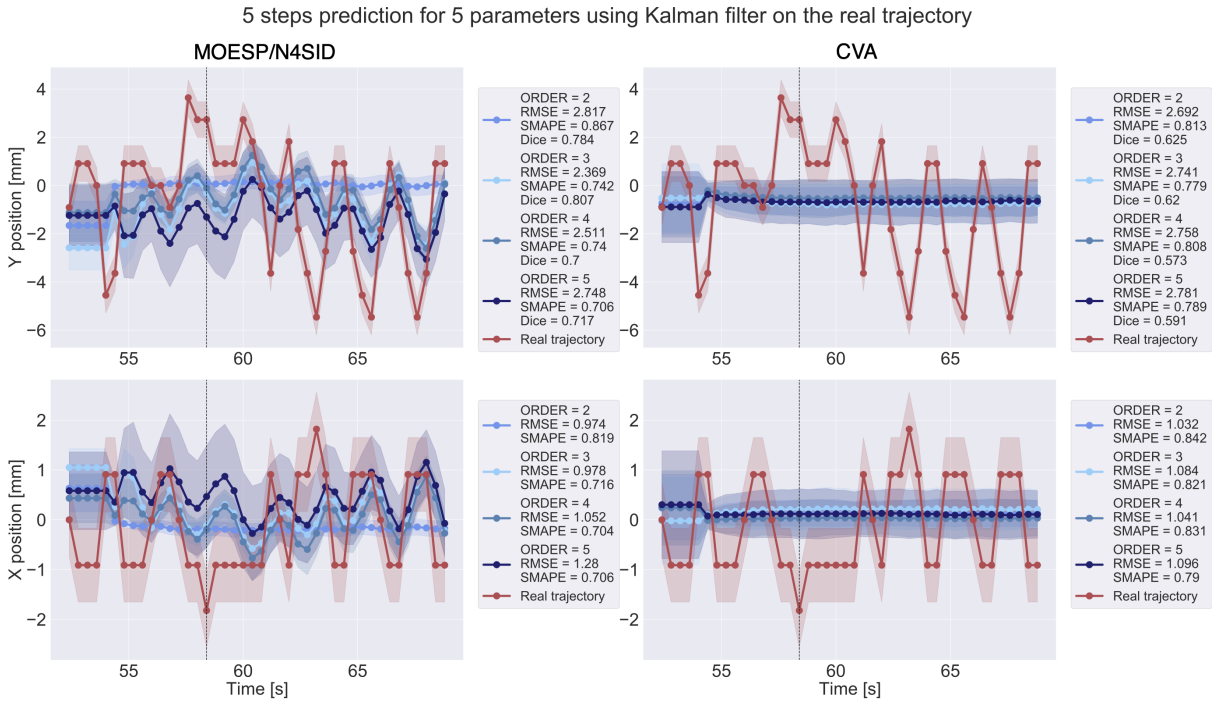

Figure 5.25: 5 steps prediction for 5 parameters using Kalman filter on the real trajectory. The mean real trajectory in red, the measurements in light red and the prediction in blue. The model is found with a system identification and three identification methods are used. Left: MOESP/N4SID, Right: CVA. P1C. Evaluation set. 50 simulations.

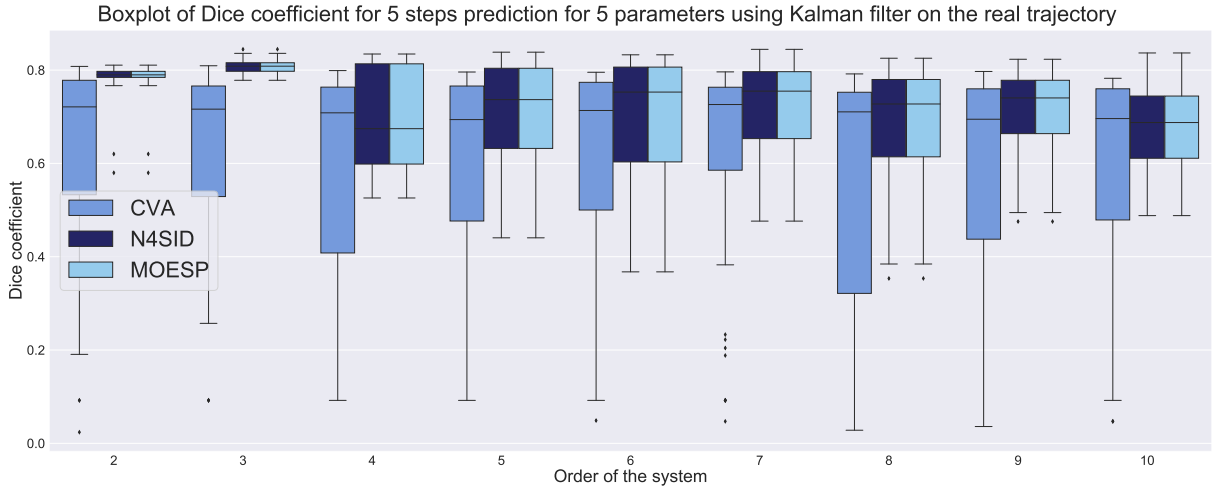

Figure 5.26: Boxplot of Dice coefficient for 5 steps prediction for 5 parameters using Kalman filter on the real trajectory. The model is found with a system identification using MOESP, N4SID and CVA. The order of the system varies from 2 to 10. P1C. Evaluation set. 50 simulations.

The best order and its performance can be found for all other Patients in Table 5.4.

		Y position		X position	
Patient	Optimal order	RMSE [mm]	SMAPE	RMSE [mm]	SMAPE
P1C	3	2.485	0.767	1.02	0.748
P1S	3	2.094	0.639	1.07	0.471
P2S1	3	1.503	0.662	0.54	0.558
P2S2	4	1.231	0.869	0.68	0.872
P2S3	3	2.124	0.541	0.5	0.452
P3C	3	4.903	0.73	1.565	0.559
P3S	5	5.493	0.637	3.283	0.844

Table 5.4: Table containing the optimal order for each patient, and the corresponding RMSE and SMAPE.

When predicting the five parameters (cf. Figure 5.26), we observe that CVA gives nearly a constant as result (cf. Figure 5.25). However, when predicting only the tumor center of mass, CVA approach gives results other than a constant. In addition, the standard deviation of the CVA is larger (due to this constant prediction) and the mean is less good than both other methods (N4SID or MOESP).
Again, N4SID and MOESP give the same results and have a very small standard deviation and a high mean Dice coefficient for the system orders 2 and 3.

In Table 5.4, the optimal order varies between 3 and 5 and the SMAPE varies between the 0.45 and 0.85.

We could think that the constant response of the CVA approach is due to the order of the system that is not high enough but even when the order equals 5, the result is a constant. To test with a higher order, a larger dataset should be used as explained in Section 2.2. Moreover, since CVA approach gives results other than a constant when predicting only the tumor center of mass, this strange observation for the prediction of the five parameters is due to the complex dynamics that it has to model.

We can conclude that 2 and 3 are the best system orders. However, an order 2 may give a constant as result because the order is too small and the system found is thus simpler (cf. Figure 5.25). For order higher than 4, the standard deviation is higher. Based on the number of data we have and the value of Dice coefficient, we determine that the order 3 is the best one even though the predicted trajectory is not perfect as shown in Figure 5.25.

The 3 to 5 system orders that we obtain as best parameters in Table 5.4 are coherent with the number of data available that we use to determine the system matrices and what can be found in the litterature.

Different simulations of trajectory with the best order are drawn on the testing set in Figure 5.27 to determine the presence of overfitting.

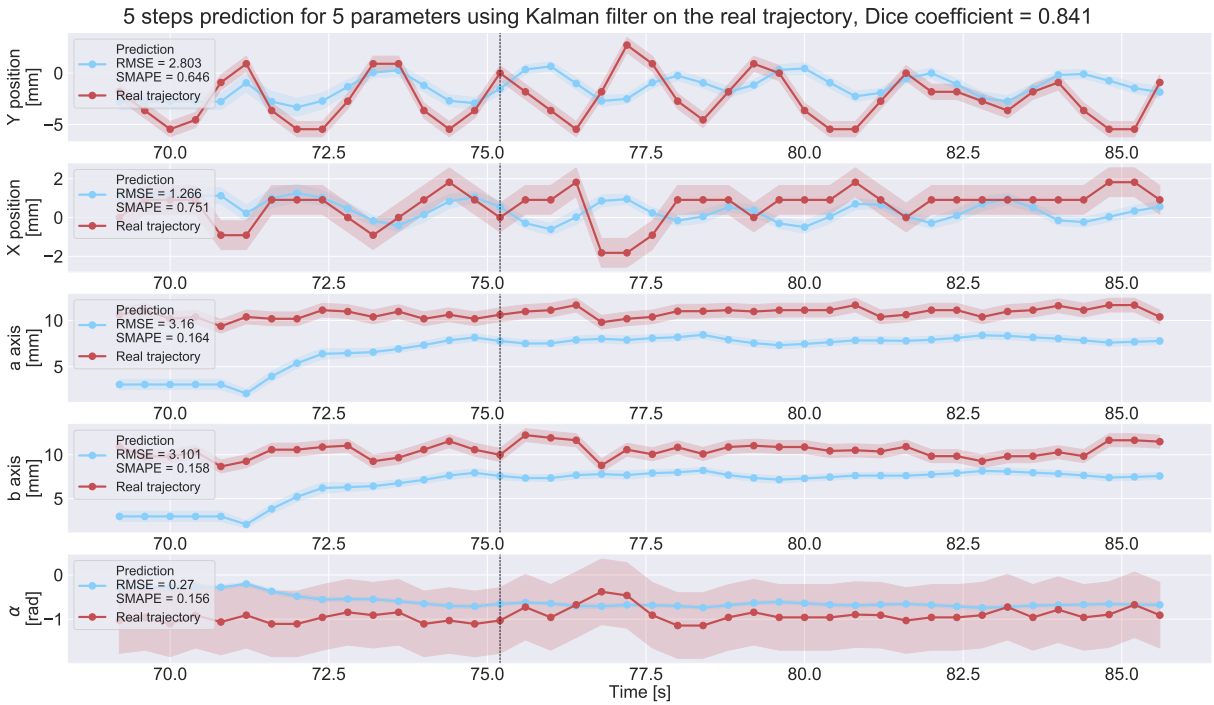

Figure 5.27: 5 steps prediction for 5 parameters using Kalman filter on the real trajectory. The mean real trajectory in red, the measurements in light red and the prediction in blue. The model is found with a system identification using N4SID/MOESP. P1C. Testing set. 100 simulations.

Again, as we can see in Figure 5.29, the prediction is not always containing the whole tumor surface. A boxplot of the Dice coefficient between the real contour and the dilated prediction with kernel sizes of 1, 3 and 5 can be seen in Figure 5.28 to determine for each patient whether a dilated predicted ellipse found is a better prediction.

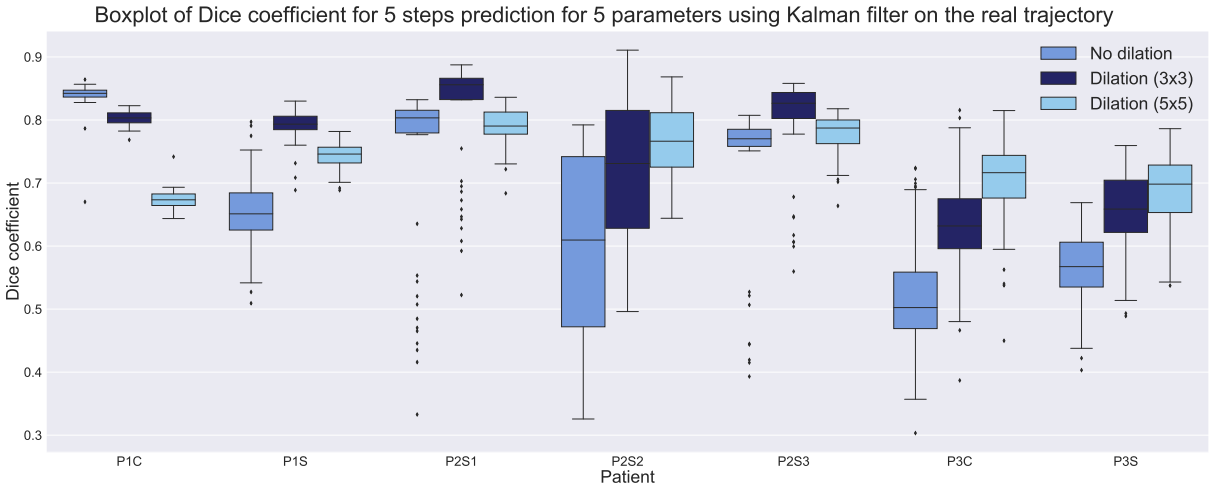

Figure 5.28: Boxplot of Dice coefficient for 5 steps prediction for 5 parameters using Kalman filter on the real trajectory. The model is found with a system identification. The dilation kernel applied on the predicted ellipse varies between 1×1, 3×3 and 5×5. Optimal parameters are used. All patients. Testing set. 100 simulations.

The following figure shows the representation of the predicted ellipse using Kalman algorithm at time $t = 81.6$ s in the testing set for P1C and P1S. On these, the predictions are represented in a non dilated form, dilated with a kernel 3×3 and 5×5.

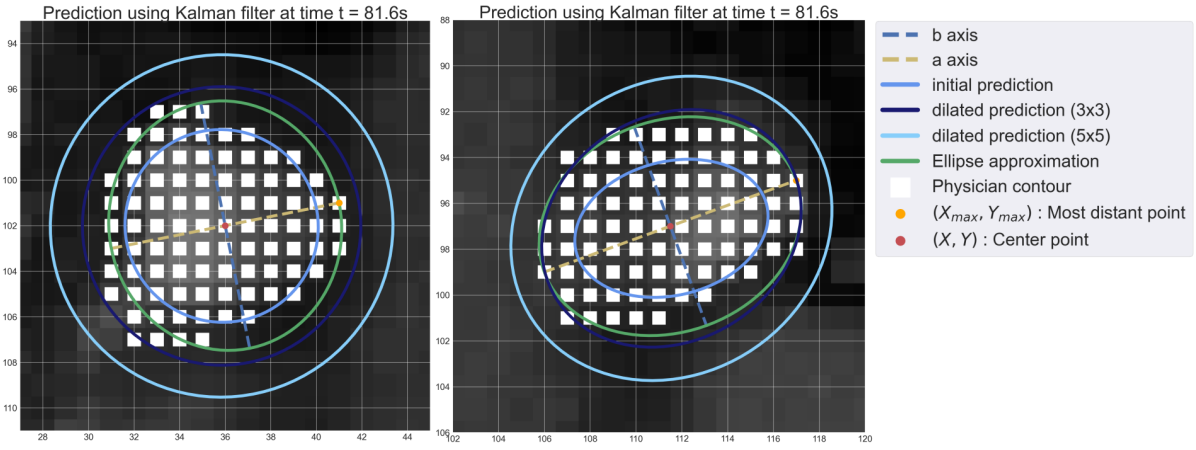

Figure 5.29: Prediction using Kalman filter. Three dilation kernels are represented: 1×1, 3×3 and 5×5. Left: P1C, Right: P1S at $t = 81.6$ s.

As we can observe in Figure 5.27, the prediction needs a certain time to correspond to the real trajectory for the a and b axes. Moreover, we see that the predictions for the X and Y positions evolve in a sinusoidal way and therefore hardly take into account irregularities occuring in the real trajectory.

The SMAPE are higher when using an identification toolbox to identify the system compared to finding it analytically (cf. Figure 5.20). The errors on X and Y positions are higher when predicting 5 parameters than when predicting only those two parameters (cf. Table 5.6).

For the model with the five parameters found with the system identification, the errors are a bit larger on the testing set than on the training set (RMSE Y = 2.37 vs 2.8 mm, RMSE X = 0.98 vs 1.27 mm) but the SMAPE remains 0.7 on both sets.

For the optimal dilation kernel (cf. Figure 5.28), the pattern is not the same for all patients but for most of them, a 3×3 or 5×5 kernel gives the best Dice coefficient. Indeed, when observing the Figure 5.29, the predicted ellipse, which is not dilated, does not comprise the whole tumor surface. It is thus needed to dilate it.

The long period of convergence for a and b axes could be due to the different ranges of values between the different parameters to predict. To solve the problem, we try to center the trajectory around 0 but that does not give better results. The best solution is to dilate the predicted ellipse since the value of a and b are always smaller than the real ones. This long time of convergence ressembles to the time needed for the Kalman filter to gain information, the adaptation time.

We consider the difference of RMSE between both sets not significant enough to consider overfitting.

The higher error on Y and X positions when predicting 5 parameters than when predicting only the tumor center of mass can be explained by the fact that the system will be more complex since 5 dynamics have to be caught. Moreover, the number of data available does not give the chance to get a robust 7 or 8 order system.

The performance of the Kalman filter to predict the a and b length axis are a bit disappointing. However, it still leads to higher Dice coefficients. We can deduce that if a dilation was again applied on a predicted ellipse, the Dice coefficient could be even bigger.

2.4 Comparison between prediction algorithms

Once the best parameters for each patient and method are found, the three prediction methods (baseline in green, sinus model in yellow and kalman filter in blue) are compared using the testing set. The comparison is made based on the SMAPE for X and Y positions, the RMSE for the five ellipse parameters Y, X, a, b, α and the Dice coefficient.

The three following tables summarize the general results that we obtain in the previous sections on P1C. This will allow us to have a global view of the best method for each kind of trajectory that we want to predict.

We can first compare the results of the three methods on two parameters (X, Y) on a regular trajectory in Table 5.5:

	Y position		X position	
Model	**RMSE [mm]**	**SMAPE**	**RMSE [mm]**	**SMAPE**
Baseline	2.7	0.58	1.38	0.55
Sinus	3.7	0.773	1.91	0.74
Kalman filter (A)	1.34	0.34	-	-
Kalman filter (SI)	0.68	0.24	0.59	0.44

Table 5.5: Summary of RMSE and SMAPE for the four prediction methods on regular breathing pattern for the center of mass (X, Y). P1C.

We can compare the results of the three methods for the prediction of only the tumor center of mass (X, Y) on a real trajectory, i.e. irregular trajectory, in Table 5.6:

	Y position		X position	
Model	**RMSE [mm]**	**SMAPE**	**RMSE [mm]**	**SMAPE**
Baseline	2.43	0.48	1.81	0.642
Sinus	3.2	0.573	1.84	0.642
Kalman filter (A)	2.46	0.473	1.53	0.78
Kalman filter (SI)	2.33	0.416	1.03	0.21

Table 5.6: Summary of RMSE and SMAPE for the four prediction methods on the real trajectory for the center of mass (X, Y). P1C.

A final general comparison can be made to conclude over the best methods to predict the trajectory of the 5 parameters of the ellipse. This can be found in Table 5.7:

Model	Y position **RMSE** [**mm**]	X position **RMSE** [**mm**]	a axis **RMSE** [**mm**]	b axis **RMSE** [**mm**]	α angle **RMSE** [**mm**]	Dice
Baseline	2.43	1.81	1.49	1.92	0.3	0.68
Sinus	3.2	1.84	1.41	1.74	0.29	0.74
Kalman filter (A)	2.46	1.53	1.65	2.0	0.83	0.78
Kalman filter (SI)	2.8	1.27	3.16	3.10	0.27	0.84

Table 5.7: Summary of RMSE and SMAPE for the four prediction methods on real trajectory for the five parameters (X, Y, a, b, α). P1C.

We can observe in Table 5.5 that Kalman filter using the system identification has the smallest SMAPE for both positions.

Compared to the baseline and sinus models, we can state that the errors of the Kalman predictor are smaller (cf. Table 5.6).

If we look at Table 5.7, we can state that the baseline model has the smallest RMSE for quite all the parameters but the higher Dice coefficient is obtained with the Kalman predictor.

Following the results of Tables 5.5 and 5.6, the conclusion is that the Kalman filter using the system identification is the best prediction model for the regular breathing pattern as well as for real trajectory for 2 parameters. Indeed, with the system found using a system identification toolbox, it provides the smallest RMSE and SMAPE for both X and Y positions.

Based on Table 5.7 and since a transformation can be made on the ellipse (dilation), we can conclude that the Kalman filter is again the best solution to predict the position of a tumor by an ellipse.

A more detailed analysis of the comparison of the different methods are made below. The comparison using the optimal parameters is represented in Figure 5.30. This allows to determine the best prediction model for each patient.

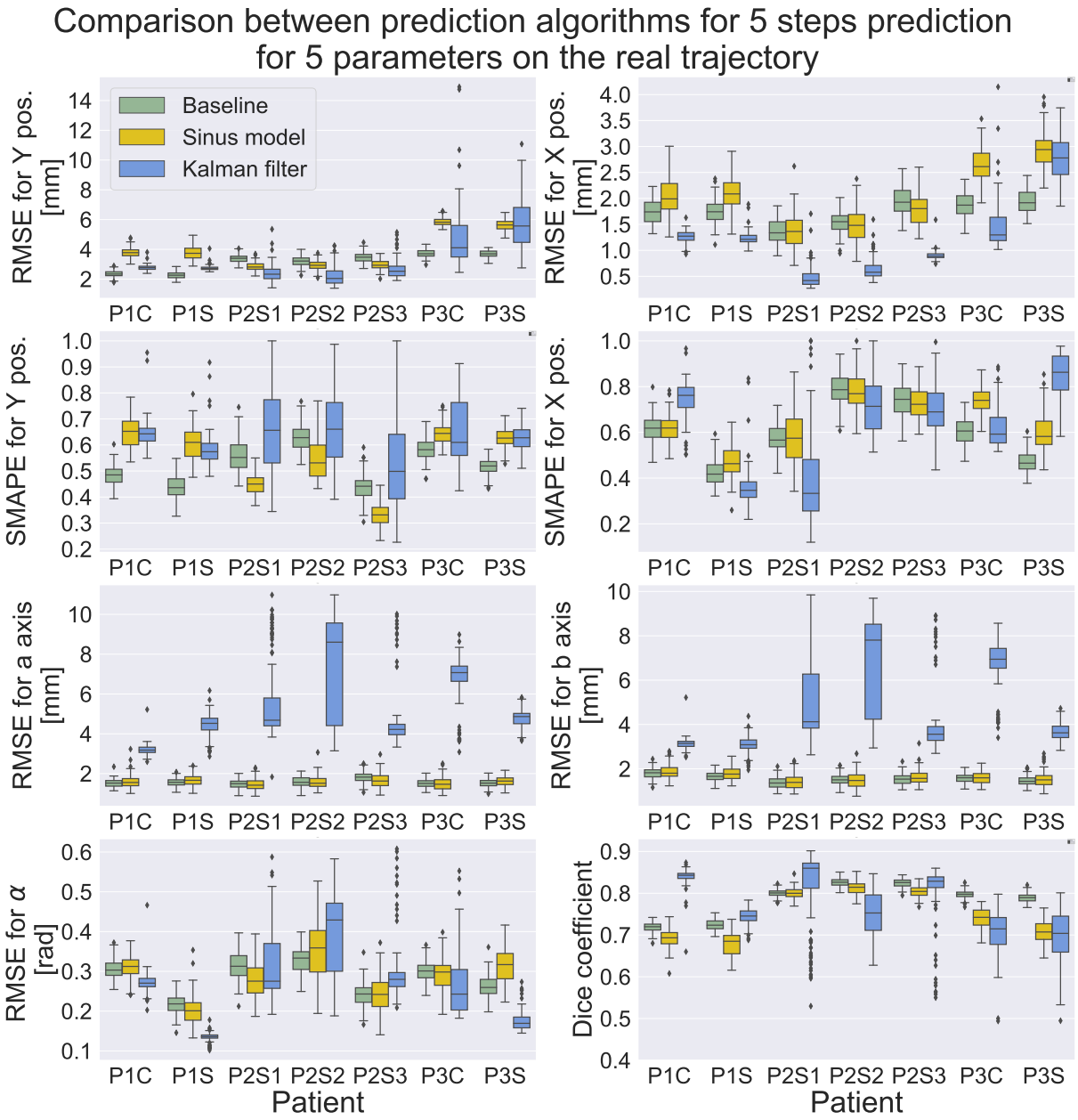

Figure 5.30: Comparison between prediction algorithms (baseline model in green, sinus model in yellow and kalman filter in blue) for 5 steps prediction for 5 parameters on the real trajectory. All patients. Testing set. 100 simulations.

After that, the prediction step is varied between 0 (no prediction) and 9 time steps. The figure 5.31 shows the performance for P1C. This figure can give us an idea about how the prediction algorithms behave following the prediction step. This allows us to determine to which time step we can accurately predict in our application but also which algorithm may be prefered following the time step at which we have to predict. The graph for the other patients can be viewed in the Annexe in Figures E.1, E.2, E.3, E.4, E.5 and E.6.

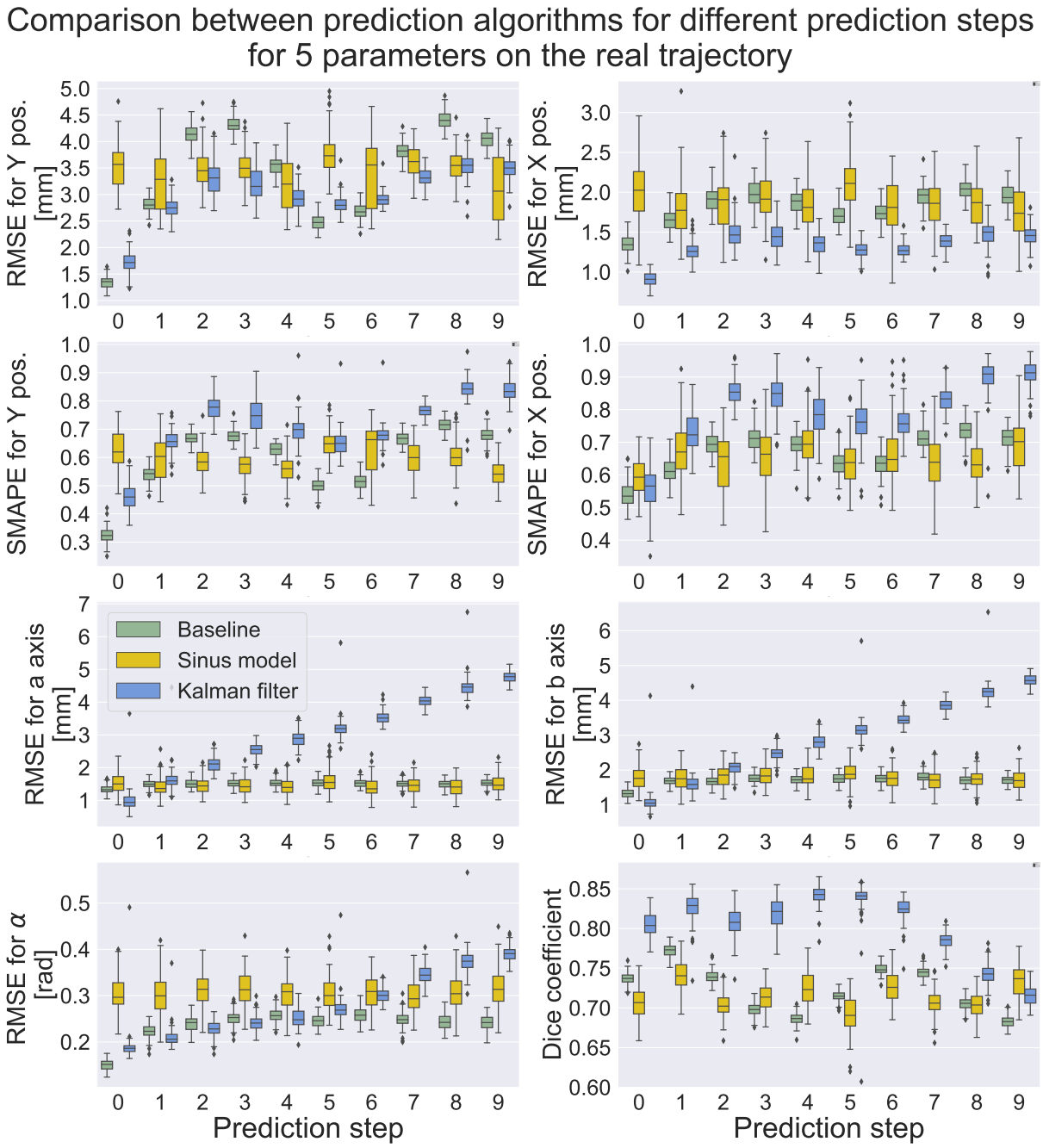

Figure 5.31: Comparison between prediction algorithms for 5 parameters on the real trajectory. The prediction step varies from 0 to 9. Optimal parameters are used. P1C. Testing set. 100 simulations.

To finish, we tried to increase the noise added on the raw data to see the influence on the prediction. The variation of the noise standard deviation goes from 0.0 to 4.4. This can be seen in Figure 5.32 for P1C. Those results for the other patients can be visualized in the Annexe in Figures E.7, E.8, E.9, E.10, E.11 and E.12. This enables to determine the robustness to noise of each algorithm and draw conclusion on which algorithm may be preferred following the noise we have in our application.

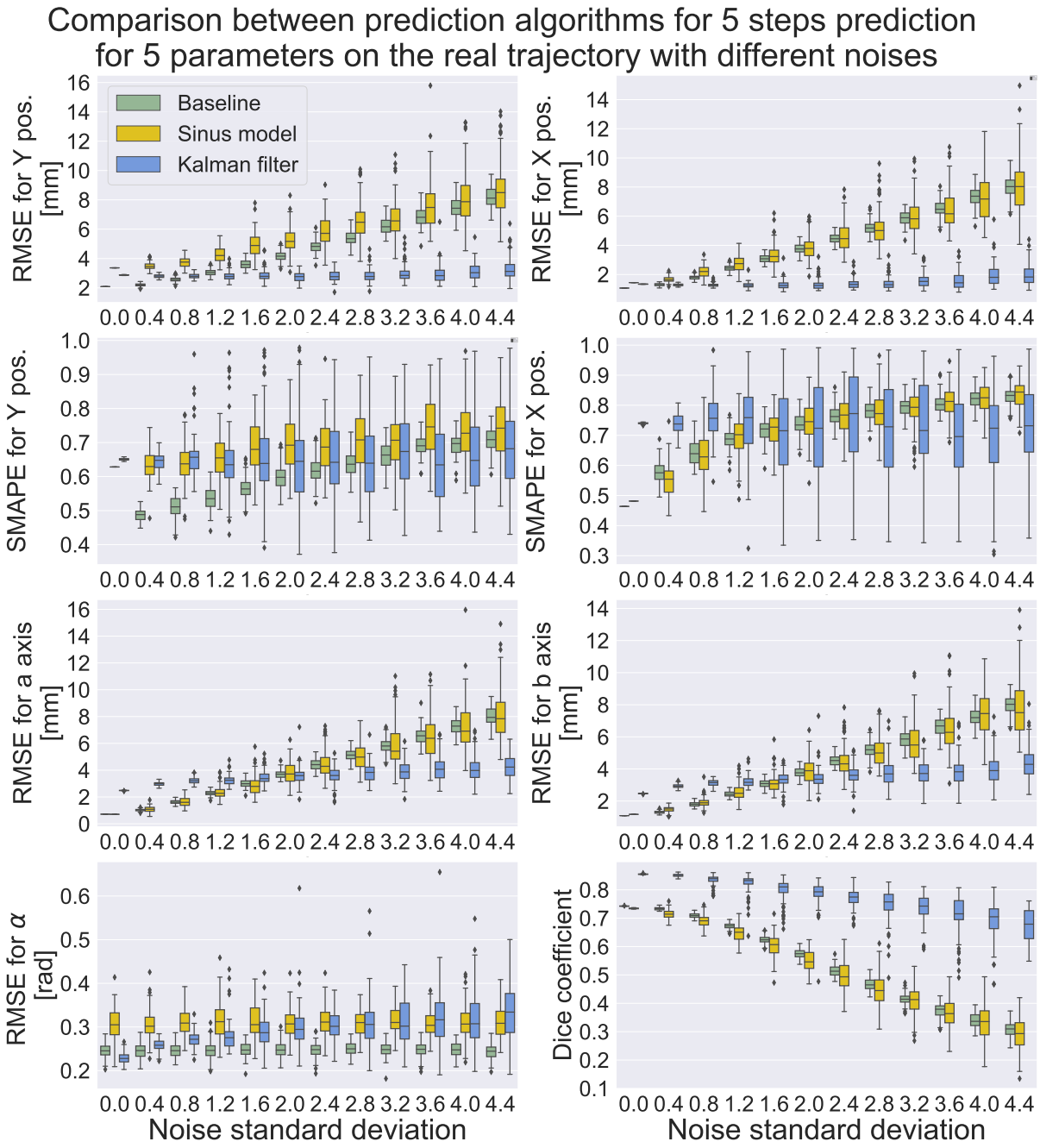

Figure 5.32: Comparison between prediction algorithms for 5 steps prediction for 5 parameters on the real trajectory. The noise varies from 0.0 to 4.4. Optimal parameters are used. P1C. Testing set. 100 simulations.

We can see in Figure 5.30 that the Kalman filter has a lower RMSE for the Y and X position for the majority of patients. If the Kalman filter has not the lowest RMSE, it is the baseline model. For the a and b axes, the RMSE of the Kalman filter is higher. This is not surprising as we saw that the prediction takes time before corresponding to the real trajectory. The RMSE of α is lower for the Kalman filter except for P2. Regarding the Dice coefficient of the Kalman filter, it is higher for all patients except P2S2, P3C and P3S. For these patients, the baseline model has a higher Dice.

In Figure 5.31, we can observe that the Dice coefficent of the baseline model varies in a sinusoidal shape with the prediction step. The performances of the sinus model are very constant when we make the prediction step vary. The Kalman filter stays the best predictor until the 7th prediction step. After that, its Dice coefficient decreases to be equal to the one obtained with other predictors. This is a general conclusion for all patients. Regarding the SMAPE, we can observe that it is higher for the Kalman filter while the RMSE is lower.

In Figure 5.32, we can see that the Dice coefficient decreases with noise for the three methods but the decrease is less pronounced for the Kalman filter. The same observation can be made for the RMSE. It increases with the noise and the increase of the Kalman filter is very small or even constant. However, the standard deviation of the SMAPE for the Kalman predictor is much bigger than for the other predictors.

For the majority of patients, Kalman predictor gives the best predicted ellipse, followed by the baseline model except for P3. The good performance of the baseline is explained by the relation between the prediction step and the period of the signal as already explained.

The sinusoidal shape of the baseline model Dice coefficient when the prediction step is varied seems logical with the observation made before. The error of the sinus prediction is constant because if the sinus model has the good parameters, predicting two, five or nine steps further only changes the moment at which we arrive in the signal but does not change the quality of the prediction model. Indeed, the model does not change with the prediction step.

Moreover, the decrease of performance of the Kalman filter after the 7th prediction step is due to the continuous increase of the RMSE for a, b and α as the step prediction increases. A dilation of more than 5×5 can be considered to solve this problem. The fact that, for Kalman, the SMAPE is higher while the RMSE is lower can be explained by the difference in terms of SMAPE of the under- and over-predicted values.

The high SMAPE standard deviation for the Kalman predictor can be explained by the very variable predicted trajectory. However, unlike the other method, the amplitude remains in the same range than the real one explaining the better performance. Indeed, the two other methods are more governed by noise.

As explained in Section 5 of Chapter 4, we also compare the prediction algorithms with a treatment without prediction, the ITV-based PTV margins. This comparison is based on two values: the proportion of tumor surface irradiated and the surface of healthy cells irradiated. The Figure 5.33 shows the results when using the optimal parameters for each patient and method. Because it is important for the treatment to cover the entire tumor with the particle beam, we tried to dilate the predicted ellipses until the proportion of tumor surface irradiated with our prediction methods is equal to 1 (cf. Figure 5.34). This allows to determine what we gain in term of surface of healhty cells irradiated by using a certain prediction algorithm instead of what is actually used (ITV-based PTV margins). These two figures allow thus to know which treatment is the best one.

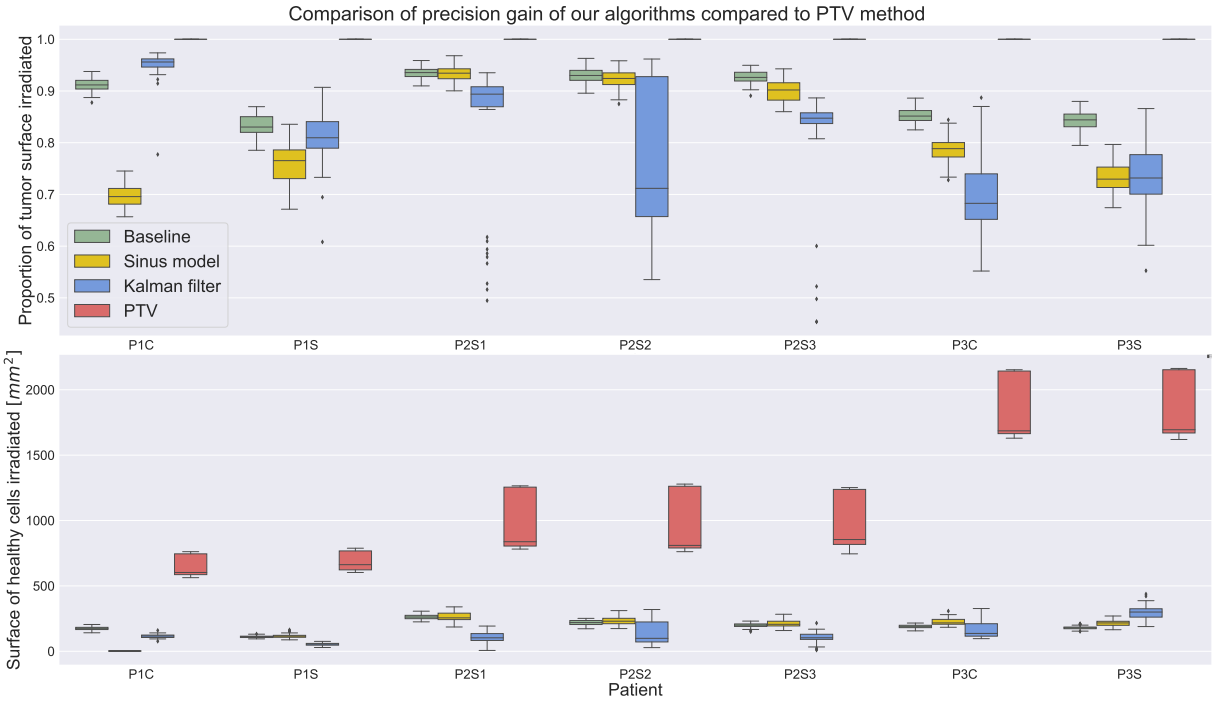

Figure 5.33: Comparison of precision gain of our algorithms compared to ITV-based PTV method: baseline model in green, sinus model in yellow, kalman filter in blue and ITV-based PTV method in red. Optimal parameters for each method and patient are used. All patients. Testing set. 100 simulations.

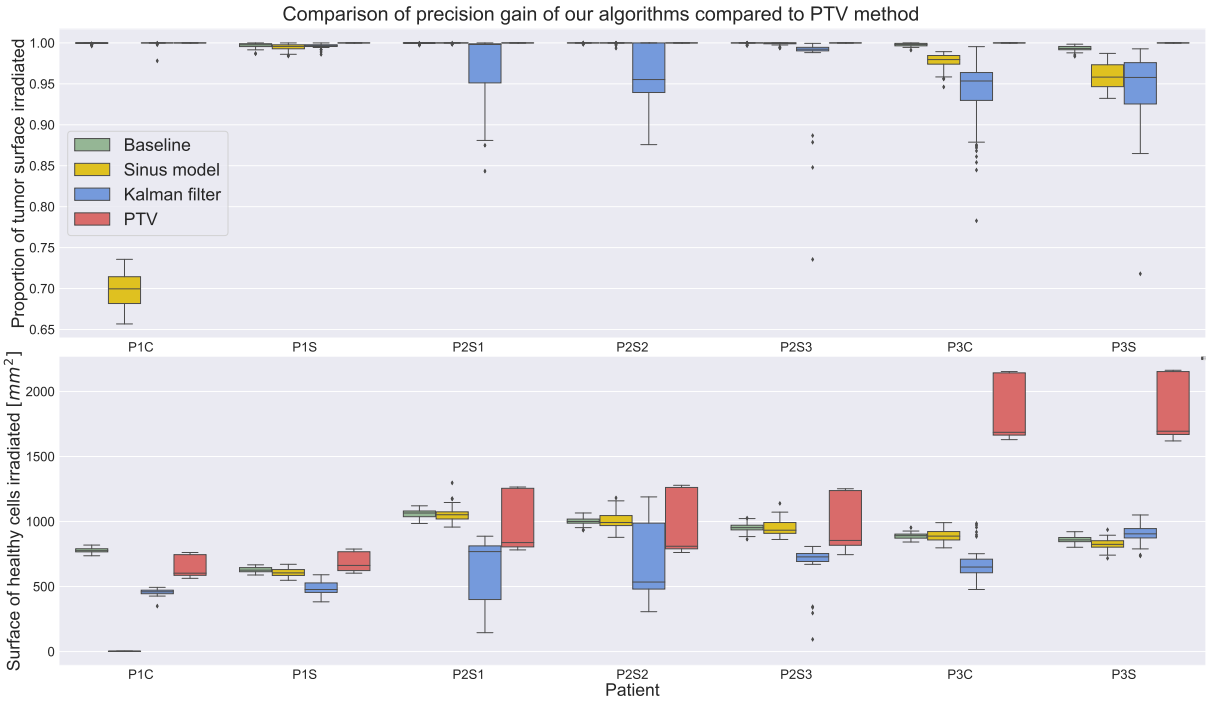

Figure 5.34: Comparison of precision gain of our algorithms compared to PTV method: baseline model in green, sinus model in yellow, kalman filter in blue and ITV-based PTV method in red. Optimal parameters for each method and patient are used. Only the kernels applied on the predicted ellipse are bigger. The kernel size is increased of 6 units. All patients. Testing set. 100 simulations.

We can see in Figure 4.5 that the tumor is entirely inside the ITV, as said by the definition in Section 3.1. This is confirmed by the Figure 5.33 where the proportion of tumor surface irradiated equals 1 for all patients when using ITV-based PTV treatment. On the contrary, the surface of healthy tissue irradiated is high compared to the prediction models tested in this work.

In terms of proportion of tumor irradiated, the same trend can be observed for P2 and P3. We can enumerate the methods from the smaller to the larger tumor surface proportion covered: the Kalman filter, the sinus model and the baseline model. For P1, the proportion of tumor surface irradiated with the Kalman filter is higher than the two other predictors. We can also observe that this proportion is more variable for the Kalman filter as the standard deviation is higher. The same tendency is shown for the surface of healthy cells irradiated. When using the contour found with the Kalman filter, less healthy cells are irradiated. One must specify that for this gain measurement, we want to have the proportion of tumor surface irradiated as big as possible (=1) and the smallest surface of healthy cells irradiated (near 0).

When we increase again the dilation kernel applied on the contour found (cf. Figure 5.34), we enlarge the predicted ellipse. As we see that the tumor surface irradiated is not equal to 1, we suppose that the dilated predicted ellipse will comprise more tumor cells. By doing this, the majority of the tumor surface is irradiated which is what is searched for the treatment. Regarding the healthy cells, we observe that it is the contour found with the Kalman filter that irradiates them the least for almost all patients.

The fact that the ITV-based PTV treatment has a high surface of healthy tissues irradiated is not surprising as the PTV is much larger than the tumor motion region.
These results indicate that predicting the tumor position can help reducing the effects of treatment on healthy tissues, more precicely the Kalman filter is the one that irradiates the least healthy tissues.

However, we can see in Figure 5.34 that the mean proportion of the tumor surface irradiated with Kalman is not exactly equal to 1. We must thus keep in mind that the surface of healthy cells irradiated may increase if this proportion equals to 1.

3 Segmentation algorithms

The goal of this section is to verify that the algorithms give sufficient results when their initialization is the more precise (i.e. based on the approximated ellipse and not the predicted yet).

For each segmentation algorithm, the superposition of the contour found on the physician contour for two patients P1C and P1S is shown. The physician contour is represented in white and the contour found with the segmentation algorithm is superposed on it in a specific color. These images come from the testing set. Only one simulation is done as there is no randomness in the segmentation algorithm.

3.1 Baseline algorithm

For the baseline algorithm, we only have to determine if the contour found needs to be dilated or not to cover the entire tumor surface. This is done based on the Dice coefficient found on the training set since no other parameters need to be optimized (cf. Figure 5.35). A representation of the contour found is given in Figure 5.36.

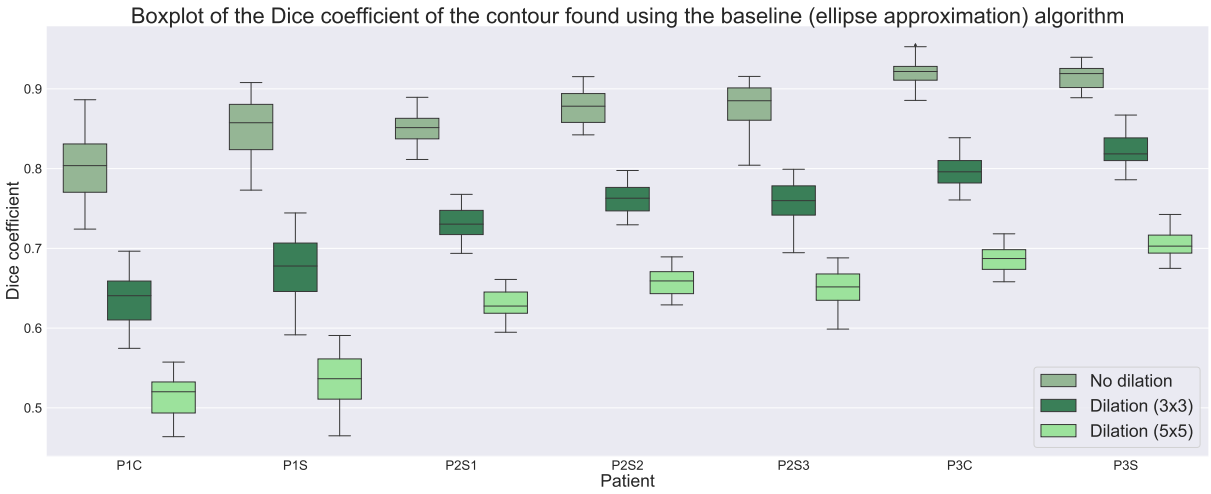

Figure 5.35: Boxplot of the Dice coefficient of the contour found using the baseline algorithm. The dilation kernel applied on the contour found varies between 1×1, 3×3 and 5×5. All patients. Training set.

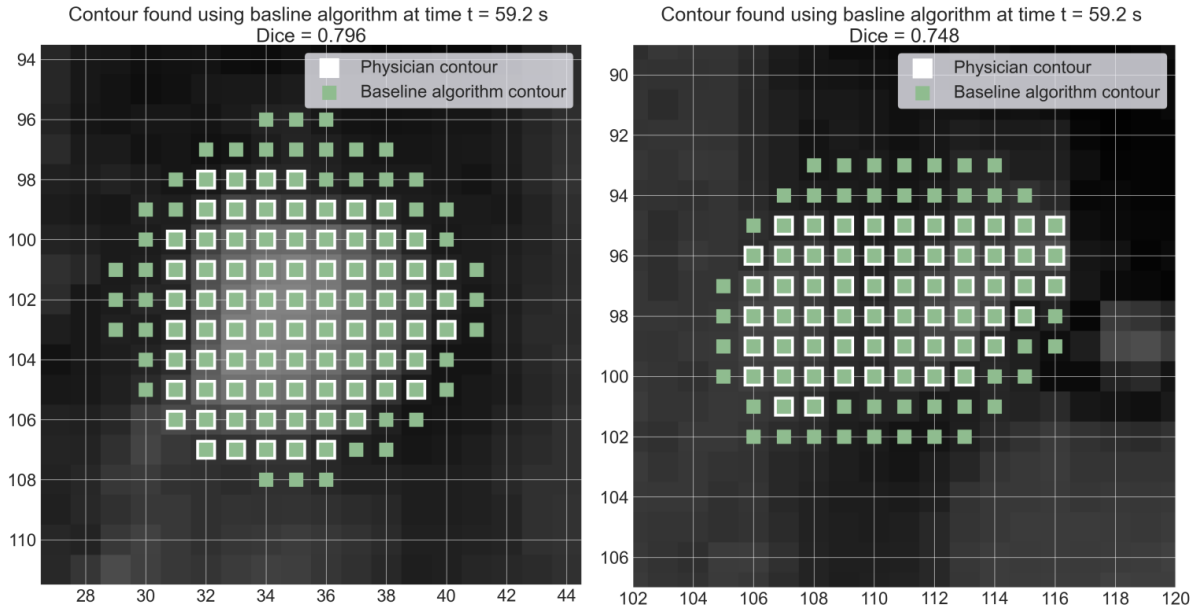

Figure 5.36: Physician contour (in white) and contour found using baseline algorithm (in green). Left: P1C, Right: P1S at $t = 59.2$ s.

We can see in Figure 5.35 that the mean Dice coefficients are higher than 0.8 for all patients. Moreover, they are always higher when no dilation is applied.

In Figure 5.36, we can see that the shape of the ellipse is a circle for P1C and oval for P1S. In addition, we can observe that the tumor is entirely contained in the ellipse and that the Dice coefficients are a bit lower than 0.8.

The high coefficients indicate that the ellipses are good approximations of the tumor.

The optimal dilation kernel is the one expected because the ellipse is directly constructed on physician contour. However, we must mention that this result can be different when combining segmentation algorithms with predictions. In this last case, the baseline model is the predicted ellipse. If the prediction is good, the Dice coefficient will be high. On the contrary, if (X, Y) is not predicted at the center of the tumor, the ellipse will only contain a part of the tumor and the Dice coefficient will be lower. If the predicted length of a and b axes is smaller than the real one, it is possible that the optimal dilation kernel is 3×3 or 5×5 because it allows to include a larger part of the tumor.

The shape of the obtained ellipses seems logic. The tumor of P1C has a roundest shape. On the contrary, the ellipse for the PS1 is more elongated. The lower Dice (< 0.8) is explained by the fact that there are some pixels which do not belong to the tumor that are included in the approximated ellipse. For our application, this is not problematic.

3.2 Region growing

For the region growing algorithm, the parameter ϵ is optimized on the training set. As reminder, ϵ is related to the condition so that the pixel is contained in the growing region. To find the optimal parameter, the value of ϵ is varied between 80% and 145%. This will be the percentage of the standard deviation of the tumor gray values taken as belonging criterium. This standard deviation is found using the five first physician contours. The appropriate dilation kernel is also determined at the same time on the training set. The optimal combination of ϵ and dilation kernel size can be determined based on the Dice coefficient, as represented for P1C in Figure 5.37. The optimal parameters for each patient are summarized in Table 5.8. Two examples of contours found with region growing algorithm are shown in Figure 5.38.

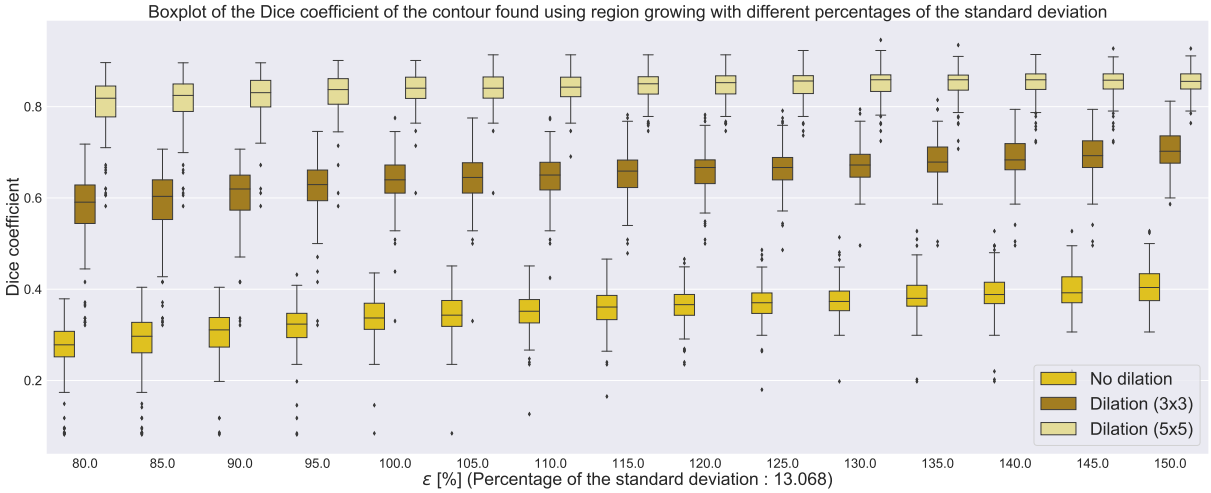

Figure 5.37: Boxplot of the Dice coefficient of the contour found using region growing algorithm. The percentage of the standard deviation (belonging parameter) varies between 80% and 145%. The dilation kernel applied on the contour found varies between 1×1, 3×3 and 5×5. P1C. Training set.

Patient	Optimal % of standard deviation	Optimal dilation kernel	Dice
P1C	115	5×5	0.853
P1S	115	5×5	0.582
P2S1	145	1×1	0.701
P2S2	105	1×1	0.748
P2S3	135	1×1	0.748
P3C	110	5×5	0.748
P3S	85	5×5	0.723

Table 5.8: Table containing optimal percentage of standard deviation and dilation kernel for each patient, and the corresponding Dice coefficient.

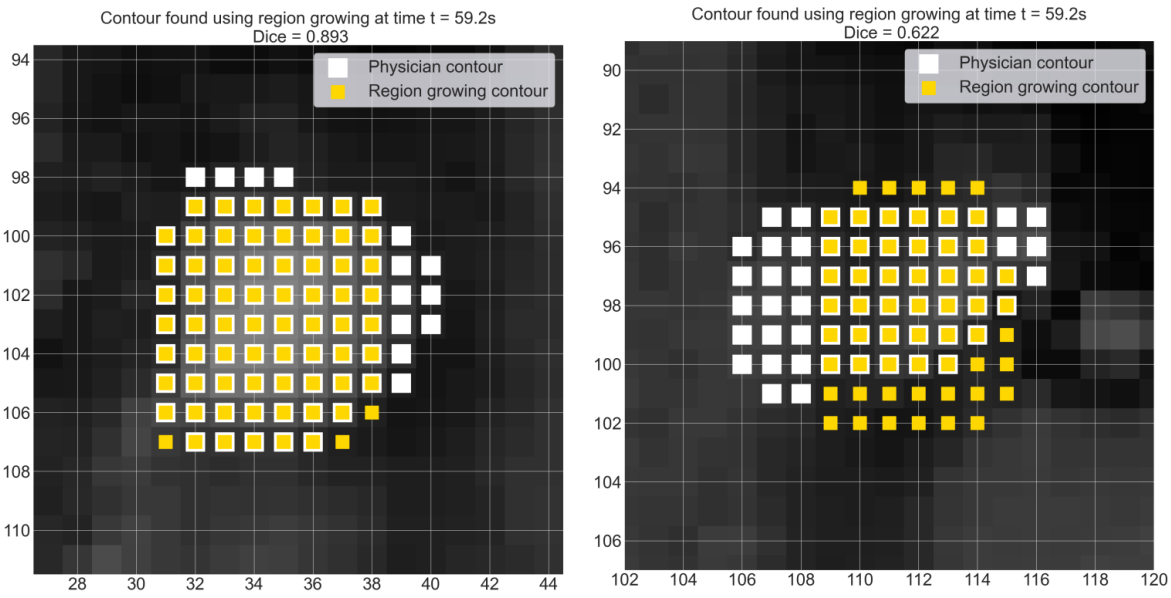

Figure 5.38: Physician contour (in white) and contour found using region growing algorithm (in yellow). Left: P1C, Right: P1S at $t = 59.2$ s.

We can observe in Figure 5.37 that there is a slight increase of the Dice coefficient when a higher percentage of the standard deviation is taken. Moreover, the Dice coefficient is better (above 0.8 for P1C) when the contour found is dilated with a 5×5 kernel.

When observing the value of the optimal percentage of the standard deviation given in Table 5.8 for the other patients, we can see that there is no specific trend.

One observation is that for all patients except P3S, this percentage is higher than 100%. Another remark is that P2 is the only patient with an optimal kernel of 1×1. Other patients have an optimal dilation kernel of 5×5.

Regarding the performances of the Region growing algorithm, the Dice coefficients are higher than 0.70 with the exception of P1S.

The proportionnality between the Dice coefficient and the standard deviation is expected. Indeed, if the belonging condition is higher, more pixels are allowed to be inside the contour until the stopping criterion is met.

The absence of trend for the optimal parameter indicates that the method is patient-specific. As reminder, we consider that sequences of different views coming from the same patient are independent.
The high percentage allows to take into account the fact that the contrast in MRI image can be low as more pixels can be included in the growing region. The values of the optimal dilation kernel can be explained by the MRI images shown in Figures 3.1 and 3.3. For P1C, there is a strong delimitation at the tumor border. The region will thus stop at the boundary while the physician contour is a bit bigger. For P1S, it is because only one part of the tumor is visible in the image and the algorithm only finds this part because it is stopped at the separation. The pixels of the other part have thus a different intensity and cannot be integrated to the growing region. As shown in Figure 5.38, only the right part of the tumor belongs to the final region. For P3, this can be explained by the pixels of a very different intensity next to each other inside the tumor itself. The segmentation algorithm stops thus very early and needs to be dilated to reach a bigger surface of the tumor.

3.3 Canny edge detector

For the Canny edge detector, the parameter to optimize on the training set is the value of the strong treshold. The initial value corresponds to the mean number of pixels n that compose the physician contour of the five first images. This value is then varied between $n - 15$ and $n + 40$ by steps of 5 units. The Dice coefficient is used to evaluate the optimal strong threshold and dilation kernel size. This can be determined thanks to Figure 5.39 for P1C. The optimal parameters and the corresponding Dice coefficient for the remaining patients are summarized in Table 5.9.

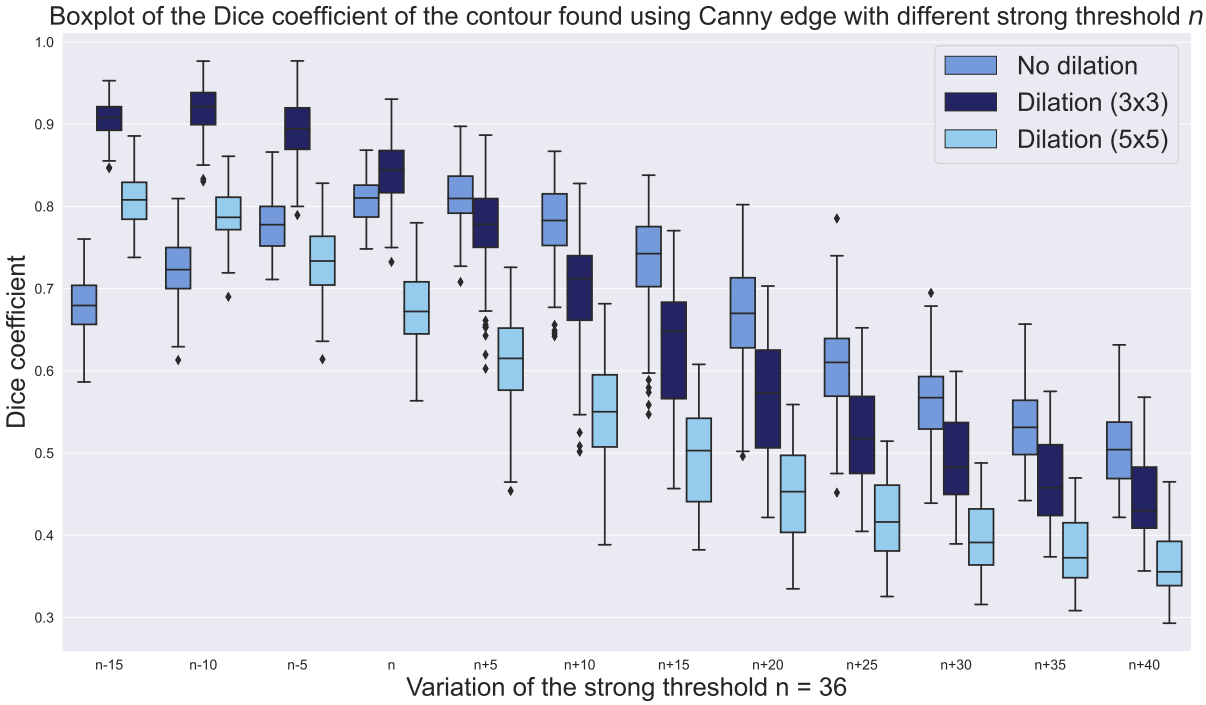

Figure 5.39: Boxplot of the Dice coefficient of the contour found using Canny edge algorithm. The strong threshold varies between $n - 10$ and $n + 40$. The dilation kernel applied on the contour found varies between 1×1, 3×3 and 5×5. P1C. Training set.

Patient	Optimal strong threshold	Dilation kernel	Dice
P1C	$n - 10$	3×3	0.918
P1S	$n - 15$	3×3	0.627
P2S1	$n + 5$	1×1	0.851
P2S2	$n + 15$	1×1	0.775
P2S3	$n + 5$	1×1	0.836
P3C	$n + 20$	1×1	0.742
P3S	$n + 25$	1×1	0.697

Table 5.9: Table containing the optimal strong threshold and dilation kernel for each patient, and the corresponding Dice coefficient.

The contours found with Canny edge detection for the two same patients are shown in Figure 5.41 with the gradient on which the algorithm is based in Figure 5.40.

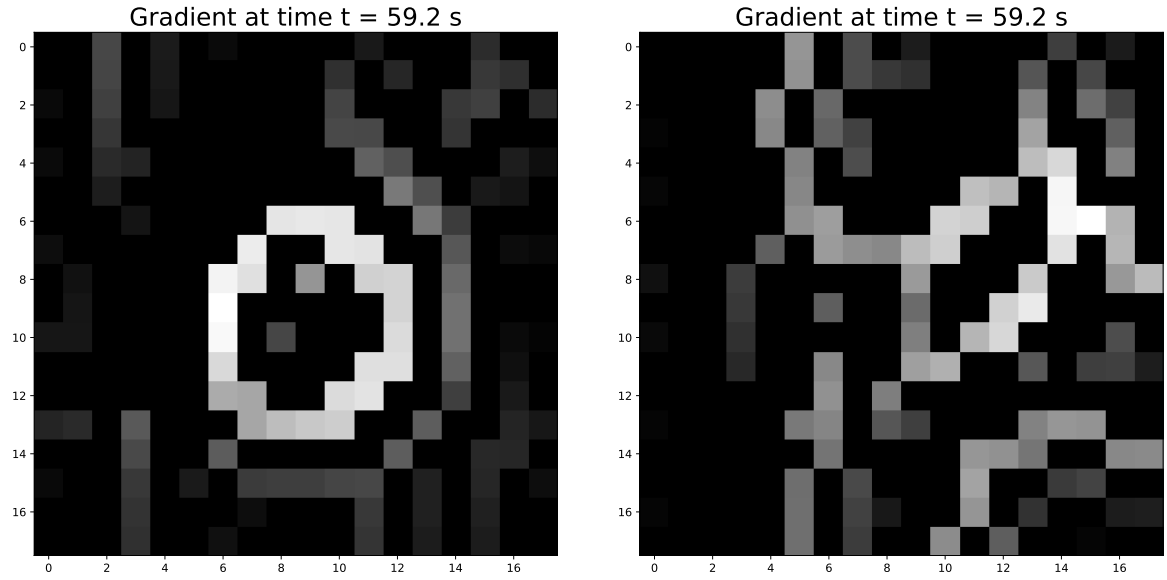

Figure 5.40: Gradient on which the algorithm was based. Left: P1C, Right: P1S at t=59.2s

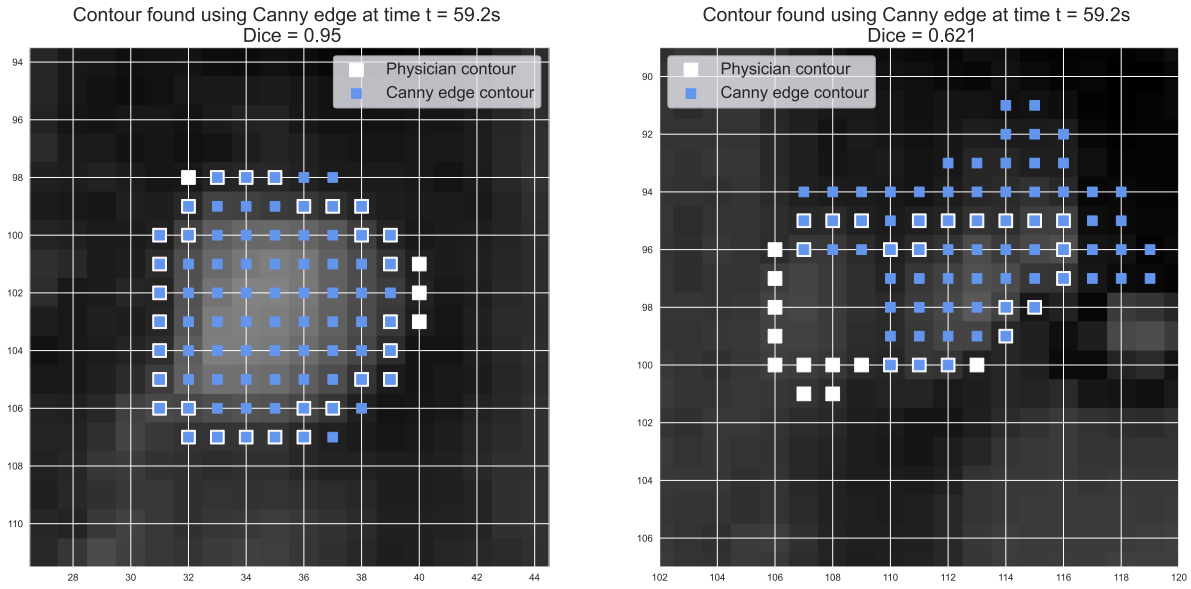

Figure 5.41: Contour found using Canny edge algorithm. Left: P1C, Right: P1S at t=59.2s

Some steps of the Canny edge algorithm are shown for P2S2 in Figure 5.42.

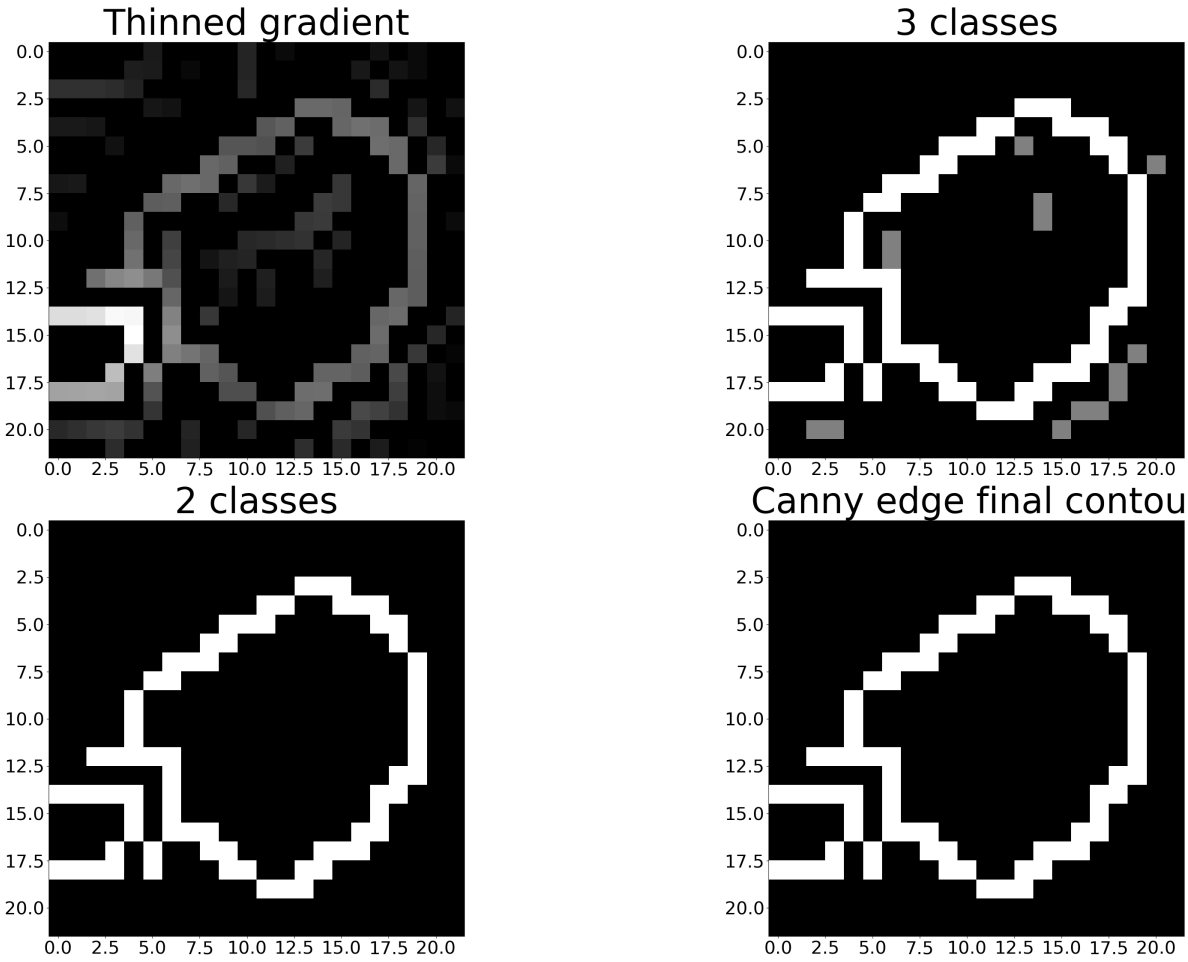

Figure 5.42: Contour found using Canny edge algorithm. P2S2 at t=0.4s

In Figure 5.39, we can see that the Dice coefficient varies when the strong threshold increases. We also see that the strong threshold equal to $n - 10$ maximizes the Dice coefficient when a 3×3 dilation kernel is applied and its mean value is around 0.9. We can observe a higher Dice coefficient with a 3×3 dilation kernel when the strong threshold has a lower value. When the strong threshold increases, the Dice coefficient is higher when a smaller dilation kernel (1×1) is applied.

When observing the value of the optimal strong threshold for each patient in Table 5.9, we can see that it is lower for P1 and higher for P3. Regarding the dilation kernel, the following observation can be conducted: the optimal dilation kernel is 3×3 for P1 and 1×1 for the remaining patients. We also see that the Dice coefficients are above 0.7 for all patients except for P1S and P3S.

The variation of the Dice coefficient with the strong threshold indicates that is has a high influence on the perfomance of Canny edge. The inverse proportionnality is explained by the fact that when the strong threshold increases, more pixels are allowed to be part of the contour and a dilation is less needed.

We can deduce a proportionality between the optimal strong threshold value and the tumor size. As given by the Table 3.1, the tumor of P1 is the smallest and the one of P3 is the largest.
The higher dilation kernel for P1 can be explained by the visibility of the tumor in the MRI images (cf. Figure 3.1) that gives a high gradient on the border as shown in Figure 5.40. The contour found corresponds to this limit. A dilation kernel of 3×3 is the best as the physician contour is always a little bit larger. The same explanation was used for the Region growing algorithm.

The reason the Dice coefficient is lower for P1S is the visibily of the tumor in the MRI image. We can indeed see in Figure 5.41 that only one part of the tumor is found by Canny edge. For P3S, the gradient does not show a very defined contour as we can see in Figure 5.43. However, the Dice coefficients have increased compared to those obtained with Region growing algorithm, meaning that Canny edge is more performant. A reason can be that Canny edge is based on the image gradient and not on the image intensity. It is thus more robust to image contrast and shows good performances even though a clear contour of the tumor is less visible in the MRI image.

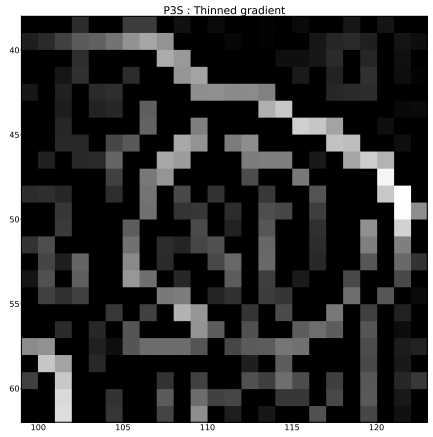

Figure 5.43: Illustration of the thinned gradient of P3S.

However, a disadvantage of Canny egde is that the presence of another organ or structure in the ROI can affect the contour. As shown in the bottom left of Figure 5.42, another structure is present in the gradient and becomes part of the contour.

3.4 Snake algorithm

For the Snake algorithm, the best parameters α, β and γ related to the snake energies are found on the training set. In order to gain time, these parameters are optimized independently of each other. The best value of α is first determined by varying its value between 0.0 and 1.0 and maintaining β and γ constant. It is varied between those values because these parameters indicate a proportion. Then, the value of β is optimized in the same way and finally, the value of γ is found. One must not forget that we use a reduced ellipse as initialisation in this part as it was explained in Section 3.3 of Chapter 2.

We can observe how the Dice coefficient varies according the α, β and γ, respectively in Figures F.1, F.2 and 5.44, to select the optimal parameters.

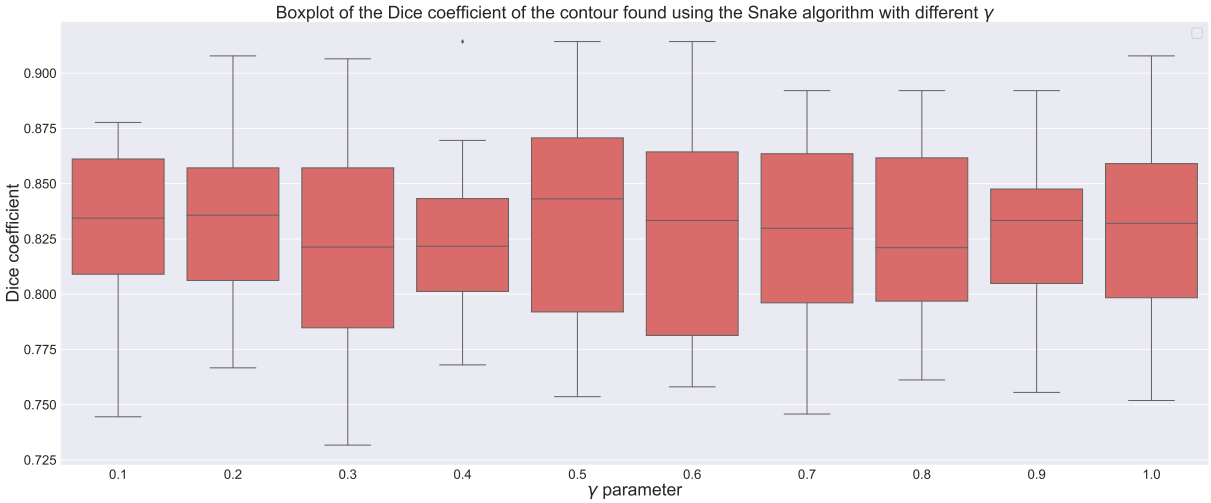

Figure 5.44: Boxplot of the Dice coefficient of the contour found using the Snake algorithm with different γ. The parameter γ varies between 0.1 and 1.0; α and β are put at their optimal value. P1C. Training set.

Once the optimal values are found, the best dilation kernel is identified on the evaluation set (cf. Figure 5.45).

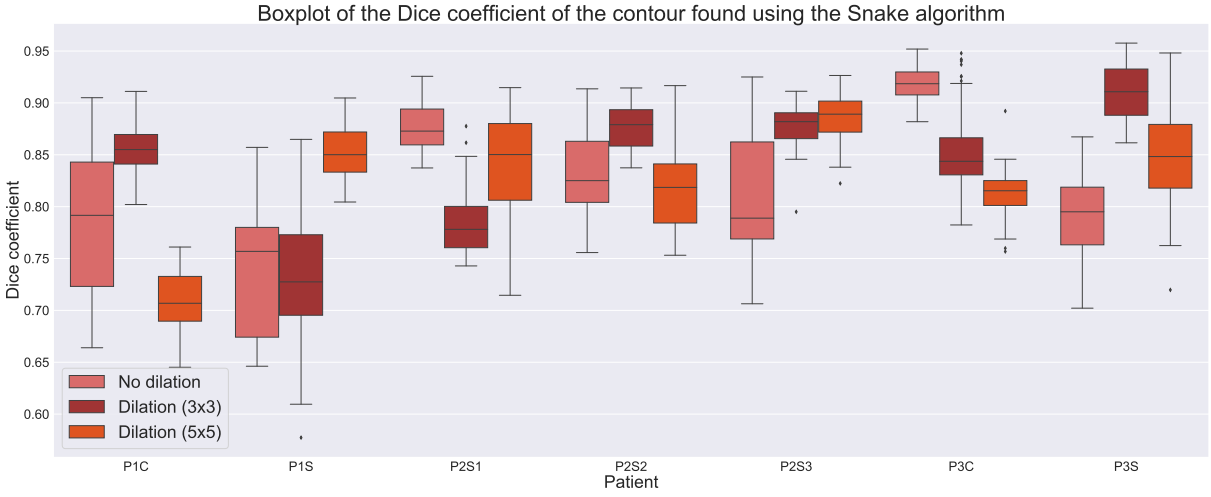

Figure 5.45: Boxplot of the Dice coefficient of the contour found using the Snake algorithm. The dilation kernel applied on the contour found varies between 1×1, 3×3 and 5×5. Optimal parameters are used. All patients. Evaluation set.

A summary of the optimal parameters for all patients is given in Table 5.10.

Patient	Optimal α	Optimal β	Optimal γ	Dilation kernel	Dice coefficient
P1C	0.1	0.2	0.5	3×3	0.895
P1S	0.1	0.2	0.5	5×5	0.851
P2S1	0.3	0.3	0.5	1×1	0.900
P2S2	0.6	0.3	0.5	3×3	0.918
P2S3	0.5	0.2	0.6	5×5	0.928
P3C	0.2	0.1	0.5	1×1	0.92
P3S	0.3	0.2	0.5	3×3	0.912

Table 5.10: Table containing the optimal parameters α, β, γ and dilation kernel for each patient, and the corresponding Dice coefficient.

Again, contours found with the Snake algorithm for P1C and P1S at $t = 59.2$ s are given in Figure 5.46.

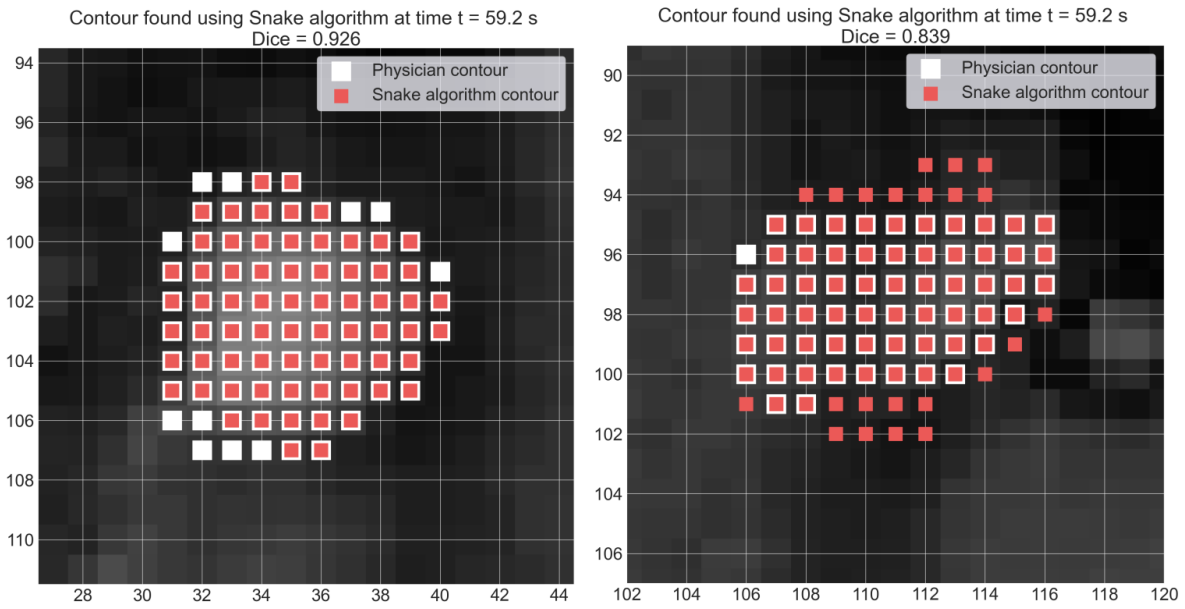

Figure 5.46: Physician contour (in white) and contour found using Snake algorithm (in red). Left: P1C, Right: P1S at $t = 59.2$ s.

In addition, two examples of the initial snake with the final snake can be visualized in Figure 5.47 and 5.48 to highlight some specific behaviour of this algorithm. The final contour for P1C and P3C is also placed onto the physician contour to be compared visually.

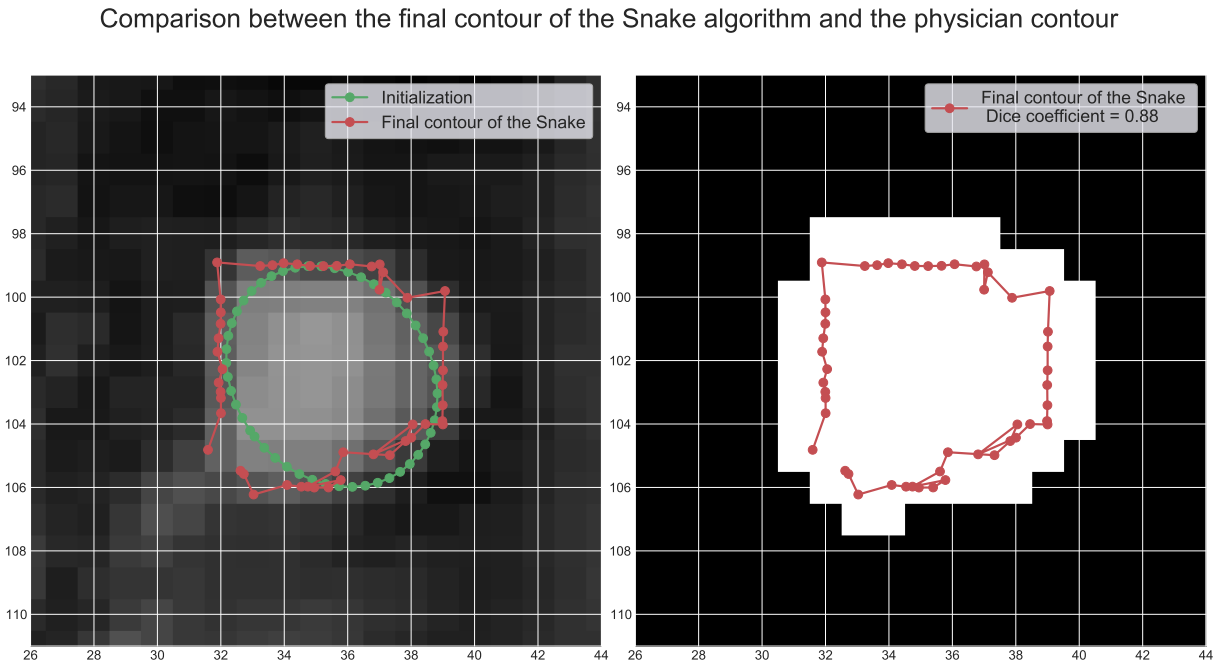

Figure 5.47: Comparison between the physician contour (in white) and the contour found with the snake algorithm (in red). The parameters used are: $\alpha = 0.9$ and $\beta = 0.9$ and $\gamma = 0.5$. The initial ellipse (in green) is reduced. Left: MRI image, Right: physician contour. P1C at $t = 1.2$ s.

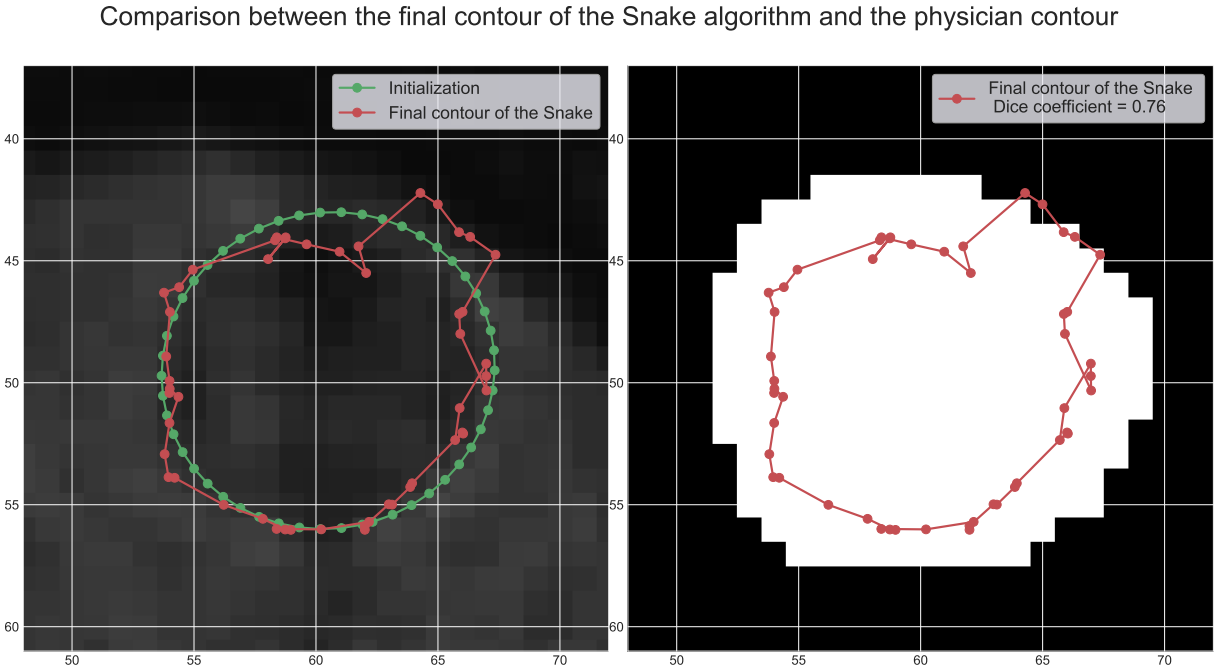

Figure 5.48: Comparison between the physician contour (in white) and the contour found with the snake algorithm (in red). The parameters used are: $\alpha = 0.9$ and $\beta = 0.9$ and $\gamma = 0.5$. The intial ellipse (in green) is reduced. Left: MRI image, Right: physician contour. P3C at $t = 1.2$ s.

In Figure F.1, we observe a slight variation of the Dice coefficient with the parameter α. All means are above 0.8 which is quite good. It has a higher effect on the standard deviation of the Dice coefficient.
The influence of the parameter β on the mean and standard deviation of the Dice coefficient is the same as the parameter α: a low effect on the mean and a high effect on the standard deviation as shown in Figure F.2. The Dice coefficient remains above 0.8 for each β value.
In Figure 5.44, a higher variation of the standard deviation of the Dice coefficient is observed when the parameter γ varies. Again, the mean Dice coefficient is always above 0.8 with the optimal parameters on the training set.

In Table 5.10, we can see that the Dice coefficient is above 0.85 for all patients. The performances of the Snake algorithm are thus higher than Region growing and Canny edge detector.

For the optimal dilation kernel given in Figure 5.45, we do not observe a specific trend. The optimal γ equals 0.5 for all patients except for the patient P2S3. The optimal β varies between 0.1 and 0.3. The optimal value for α is the most variable, we do not see any tendency.

In Figure 5.46, we can observe that the final snake corresponds very well to the tumor contour. Even though some pixels belonging to the tumor are not included in the final contour and some pixels of healthy tissue are included in it, the number of wrong pixels is lower than the one found with Region growing and Canny edge.

This conclusion can also be seen in Figures 5.47 and 5.48 on the MRI images and on the physician contour. The contour found for P1C in Figure 5.47 fits very well to the perfect tumor contour on the MRI image. However for P3C, the snake is attracted to the border of the organ but the contour stays well comprised in the physician contour that we would tend to dilate.

The parameter α and β have not a high influence on the mean Dice coefficient. For the parameter γ, it has a higher influence on the standard deviation compared to the two other parameters. These observations show that the snake is robust to the parameters. One must not forget that a best combination of parameters could maybe be found by optimizing them all together. Since the γ is taken as a mean value, the snake is reasonably attracted to the borders. Indeed, it is refered to what is in proximity in the MRI image and since we are taking a ROI, there are less risks that there will be another border in this region. Moreover, the smallest β parameter means that we highlight the character rigid of our snake since we allow to have angles for the pixel representation of the tumor. The α parameter is more specific to patient and to tumor shape.

As we can see in the illustration, the physician contour is enlarged. This explains why the contour found by the Snake algorithm must sometimes be dilated. Another case can occur as for P3C, where we would be tempted to dilate the contour found. This would add a proportion of healthy cells to treat in the upper right of the tumor. This will tend to decrease the Dice coefficient if we dilate too much the contour found.

3.5 Comparison of segmentation algorithms

After determining the optimal parameters for each algorithm and each patient, the four segmentation algorithms are compared on the testing set. The Dice coefficient is used to determine which segmentation algorithm gives the best results for each patient (cf. Figure 5.49).

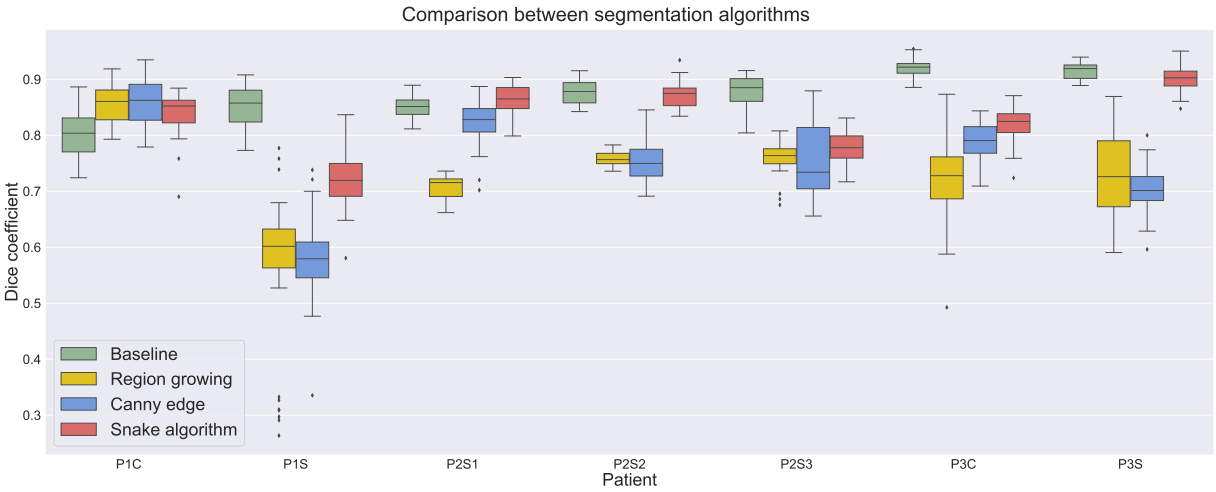

Figure 5.49: Comparison between segmentation algorithms: baseline (in green), region growing (in yellow), canny edge (in blue) and snake (in red). All patients. Testing test.

When comparing the Dice coefficients obtained with the different segmentation algorithms, a trend can be observed as shown by Figure 5.49. For all patients except P1C, the baseline algorithm is slighty more performant than the Snake with a Dice coefficient always above 0.8 while the contours found with Region growing and Canny edge present lower Dice coefficients.

We can conclude that the baseline is the more performant algorithm followed by the Snake algorithm. These two segmentation algorithms are the best for all patients except for P1C for which Region growing and Canny edge give higher Dice coefficients. Indeed, this tumor is the most visible one in MRI images. The higher performances of the baseline algorithm for other patients can be explained by the contour which is less good delimited on the images and the low number of iterations.

One can not forget that the initialisations of the algorithms are based on the physician contour and are thus perfect initialisations. The results can thus be very different if the predicted ellipses are used instead, as shown in Figures 5.50 and 5.52.

4 Combination of prediction and segmentation algorithms

The goal of this section is to combine both algorithms (of prediction and segmentation) to see the performance of the entire process. For that, it is decided to test the combination of the more performant and the more relevant prediction methods with the four segmentation algorithms. The predictions given to the segmentation algorithms are thus the one found with the Kalman filter since they give the best results. The analytical way but also the system identification model are used while all segmentation algorithms are tested. The purpose is to determine whether one combination of prediction-segmentation algorithm is more performant for all patients or if it is more patient-specific.

4.1 Kalman filter: analytical

After testing each part, we combine the prediction with the Kalman filter, whose dynamics is based on analytical model, and the segmentation algorithms. The Dice coefficients between the contour found and the real position of tumor are represented in Figure 5.50. An example of the contours found with the segmentation algorithms for P1C is shown in Figure 5.51. Those two Figures allow to conclude quantitatively and visually on the best segmentation algorithm that must be used following a prediction from the analytical Kalman filter.

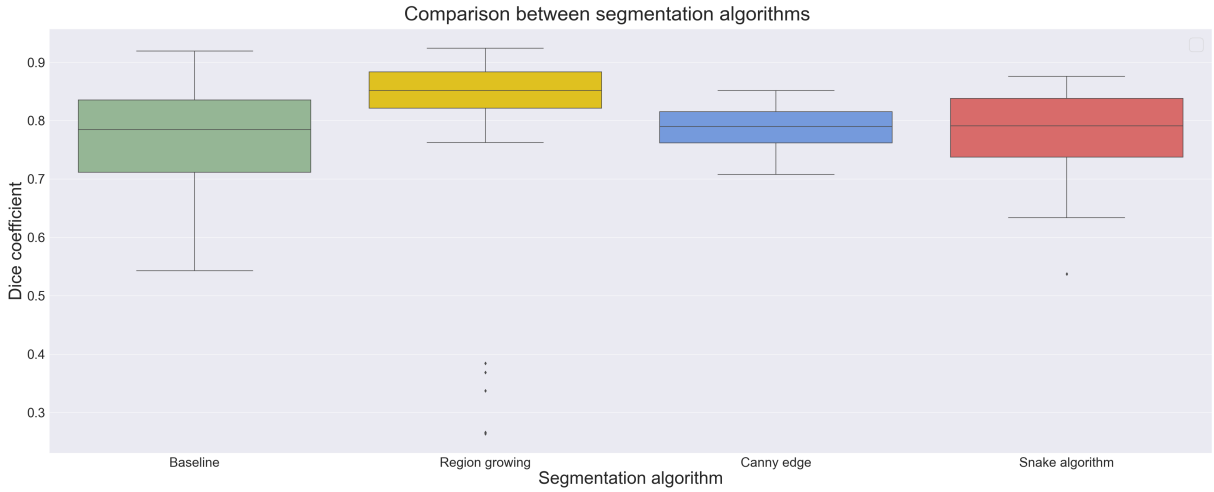

Figure 5.50: Comparison between segmentation algorithms. The predicted ellipses are obtained using the Kalman filter. The model is found analytically. P1C.

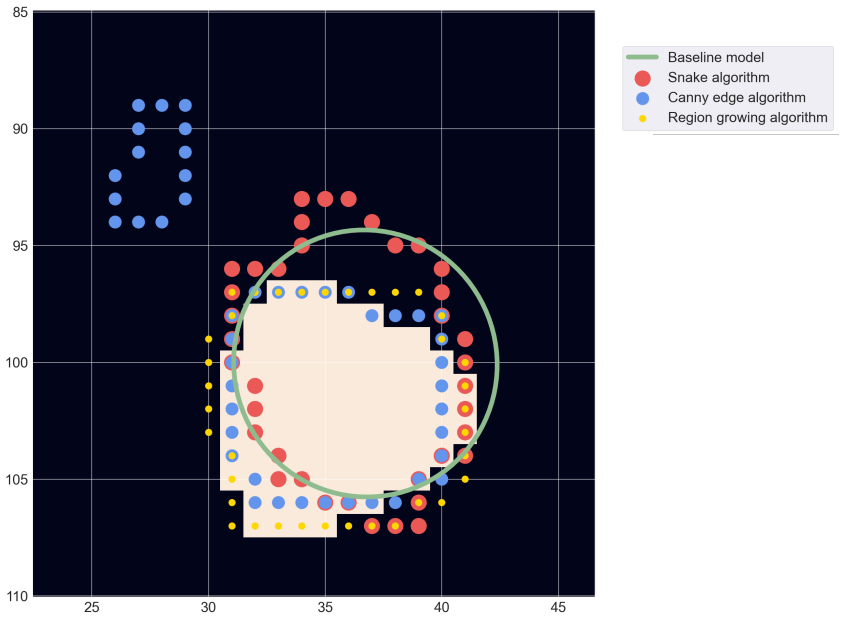

Figure 5.51: Physician contour (in white) and contours found using the segmentation algorithms. P1C at $t = 59.2$ s.

4.2 Kalman filter: system identification

We also combine the Kalman filter where the model is found with a system identification with the different segmentation algorithms. The Dice coefficients for all combinations and all patients can be found in Figures 5.52 and 5.53. This allows to determine quantitatively and visually which combination is the most performant.

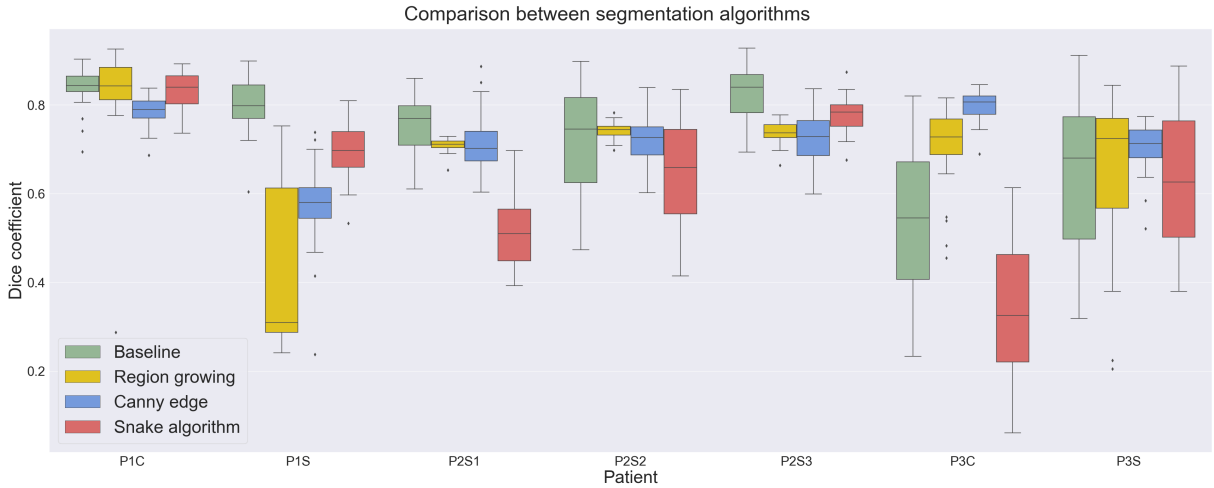

Figure 5.52: Comparison between segmentation algorithms. The predicted ellipses are obtained using the Kalman filter. The model is found with a system identification. All patients.

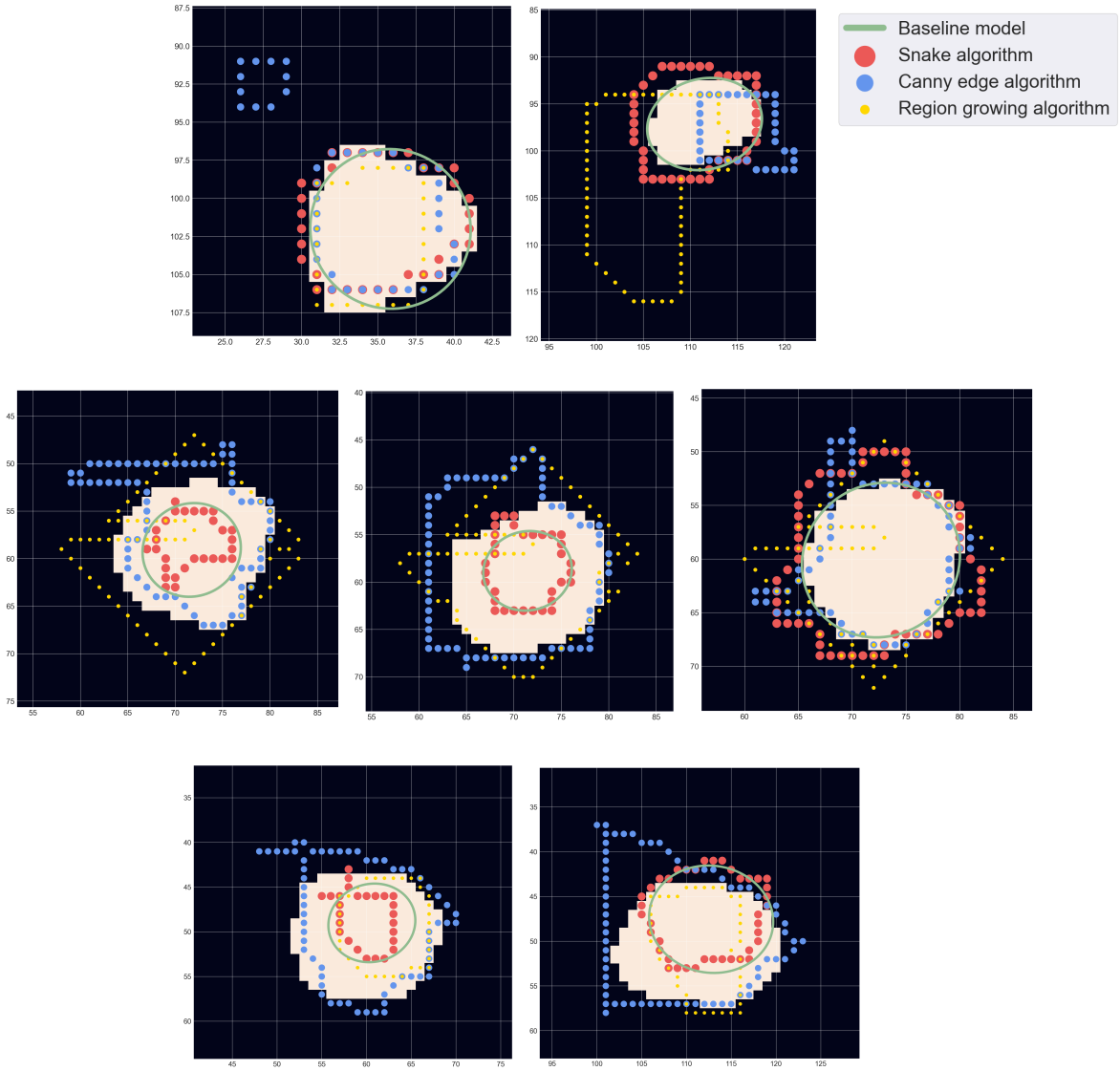

Figure 5.53: Physician contour (in white) and contours found using the segmentation algorithms. All patients at $t = 59.2$ s. From up to down and left to right: P1C, P1S, P2S1, P2S2, P2S3, P3C, P3S.

In Figure 5.50, we see that Region growing has the best Dice coefficient while it had the lowest one when using the physician contour as initialisation of the segmentation algorithm. The contours found with the three other algorithms have a similar mean Dice which is around 0.78.

Figure 5.52 shows no trend but the baseline algorithm has the best Dice coefficient for all patients except P3C and P3S. For each patient, the best method gives a mean Dice above 0.75.

Figure 5.51 shows a good example of the limitation of Canny edge. On the upper left, there is an undesirable object that is part of the contour while it should not. The contour found with Region growing includes well the tumor. The one found with the Snake is shifted compared to the tumor position. It remains around the ellipse whose center is not exactly at the center of the tumor. Perhaps with more iterations, the Snake would find the right contour.

For the contour found based on the predicted ellipse using the Kalman predicted trajectories, what we can conclude is that the baseline models gives the best results followed by the Canny edge despite some limitations due to other structures nearby for approximatively all patients. Even when the predicted ellipse is too small, the region found with the Canny edge is well enlarged so that it covers the whole tumor surface. The snake is restricted to the number of iterations for time reasons, and this is visible since it is very often not so far from the predicted ellipse. The baseline model is sometimes too small and is entirely comprised in the physician tumor contour. It is also very dependent on the prediction precision. The Region growing algorithm is working properly on tumors which have a visible contour in the MRI image as it is the case for P1C. Conclude that the method is often better without refinement seems not really precise since the ellipse is mostly comprised in the tumor (cf. Figure 5.53). The Canny detector gives often very good but bigger contour. The fact that the conclusion based on the boxplot about those two algorithms is biased is mainly due to the definition of the Dice coefficient.

However, the mean Dice of the best segmentation algorithm for all patients except P3S is above 0.78. Following what was found in the literature, these are satisfying results.

To summarize, the best algorithm to track a tumor during a treatment would be the combination of the prediction with a Kalman filter based on a dynamics found with the system identification. Since the value of the a and b are often under-estimated, i.e. their predicted values are smaller than the real ones, the predicted ellipse must be dilated to better cover the tumor. This would be followed by the Canny edge detector to know precisely the tumor contour. Concerning the segmentation, this is the most stable segmentation algorithm that does in general not need to be dilated. This complete scheme lead to a quite performant tracking algorithm.

Chapter 6

Limitations and future improvements

This work represents only a track to solve the problem of mobile tumors and has limitations that remain to be improved.

The first limitation of our study about the tracking of mobile tumors is the data used. As we work in 2D images, we do not consider tumor volume and only consider the motion in the x and y directions. The movement in the z direction is thus not taken into account.

A second limitation over the data is the number of images in each MRI image sequence. As we have to divide each sequence in three different sets, each set contains far fewer images. To identify the system for the Kalman filter, the data set contains around 100 images. That limits the number of parameters that can be identified and thus the order of the system.

In addition, we must take into account that the tumor contours used as ground truth are delimited manually and it is thus subjective, i.e. depends on the person who does it. The trajectories can thus be a little more irregular than in reality.

Regarding the metric used for the evaluation of the recovery of our prediction and of the contour of the segmentation algorithms, the Dice coefficient, it could be more indicative of a complete recovery. Indeed, it would be preferable to have a metric that can be proportional to the surface of the tumor. On the one hand, it would be better for the metric to be bigger than one if the structure we get was containing the tumor. On the other hand, it would be better for the metric to be smaller than one if the structure was comprised inside the tumor contour and equal to 1 if perfectly covered. As a result, we could discriminate between contours that contain the entire tumor or not since it is important for our application.

One evaluation that we have not done and that may enrich the study is the evaluation of the system identification process. Indeed, here the well functioning of the system identification step is tested with the Kalman filter algorithm. We consider that it is indirectly tested via the error quantification of the Kalman filter. We could maybe dissociate both processes to understand which one induces the (bigger) mistakes by evaluating directly the identification process in a further work with V_n the estimated error returned by the identification toolbox as explained in Section 2.4.3 of Chapter 4.

As explained in the previous chapter, there is a problem when constructing the sinus model as the phase shift is not correclty taken into account. This is a point that could be enhanced.

The performances of the Kalman filter could be enhanced too. It could be interesting to test whether maintaining a and b axes constant improves the Dice coefficient. Especially since it is assumed that the tumor does not rotate and deform much. As we do not do it, the system is more complex. However, the Kalman filter is limited to linear systems. As the tumor movement is complex, it could be interesting to try using the extended Kalman filter or the unscented Kalman filter.

As already discussed, one limitation of Canny edge is that the performances decrease when another light structure is present inside the ROI. However, if the physician is present, he can realise that there is a problem if two contours are found.

Another limitation of Canny edge and the Snake algorithms is that we use a ROI. That can induce several problems. The first one is that the tumor can go out of the ROI if the patient suddently breathes with a higher amplitude. That also raises a problem if the center of the predicted ellipse is not inside the ROI.

All the algorithms could be optimized with a specified and more precise optimization technique like a gradient descent technique for example. Here an inverse parametrisation has been done. For example, an optimized Snake algorithm could do less iterations to find a better contour. Better combinations of parameters can be found by optimizing the parameters all together and not independently anymore. Indeed, we saw here that more iterations are needed. Unfortunately, more iterations lead to more time.

The combination of prediction and segmentation algorithms we get is not the final goal that we wanted to reach. Indeed, the goal is to use the data obtained with the segmentation algorithm as measures to predict. More precisely, after refining the prediction with a segmentation algorithm, an ellipse must be reconstructed and the value of these parameters should be used to provide the prediction algorithm and make it work no longer on data from the physician tumor contour.

This combination is more complex to achieve since both algorithms must be embedded and will be dependent on each other. This could lead to more serious errors since an error is passed on to another algorithm, which in turn can generate more and more important errors. This leads to the conclusion that our algorithm is not usable coarsly in clinics. One step further has to be done.

Another limitation which is important if used in clinics is the notion of real-time. Indeed, we do not really take into account the time needed for each algorithm to run. This could also be optimized since our algorithms were coded in an intuitive way. Our algorithms take relatively little time but this could be more performant on more powerful machines.

Chapter 7

Conclusion

To summarize this work, the following conclusions can be drawn.

Regarding the prediction models, we observed that the Kalman filter based on a dynamics given by a system identification gives the best results on a regular breathing pattern for two parameters (predicting the position of the tumor center of mass). The RMSE is around 2.3 and 1.0 mm for the Y and X positions. On the real trajectory for two parameters, the Kalman filter remains the best prediction model as the RMSE equals 2.33 and 1.03 mm for both positions. It is also the case when predicting five parameters even if the RMSE are a bit higher. The Kalman filter has the higher Dice coefficient (above 0.7) except for P2S2 and P3. It remains the best even when increasing the noise contained in the data and varying the prediction time steps. However, the RMSE is large for the a and b axes, meaning that the Dice coefficient can increase if we apply a larger dilation kernel. For both positions, the RMSE are below 3 and 1.5 mm except for P3. After the Kalman filter, it is the baseline model that shows the best performances but it is strongly dependent on the prediction step and the period of the trajectory.

One of the major conclusion is that there is a gain in terms of healthy cells irradiated when predicting the tumor compared to the use of a ITV-based PTV treatment. The minimum gain found is around 200 mm^2 of healthy cells which are not irradiated. However, to be sure that the whole tumor surface is caught by the particle beam, the predicted ellipse needs to be dilated. This can increase the performance and be useful in our application.

The segmentation algorithm which arrives best to capture the tumor region based on an approximated ellipse is the Snake algorithm (with a mean Dice coefficient of 0.85 in general with the smallest mean for one specific patient at 0.72). When combining the processes of prediction and segmentation, the Canny edge seems to be a more performant algorithm (with a mean Dice from 0.7 to 0.81 except for P1C, P1S and P2S3 where the snake takes the lead with an average of 0.83, 0.75 and 0.79). The Canny edge method has the most constant good Dice results even though it needs a physician's control for the surrounding structures that could be included in the ROI. It is the algorithm that can best adapt to the given initialization since it is the least dependent on the precision of its initialization. The Snake algorithm is too dependent on the initialization to be the best alternative based on predictions since the predictions are not 100% precise. As already said, other methods of image segmentation are effective (like Snake algorithm, Region growing) but they mainly depend on certain characteristics of the tumor and its environment. We must not forget

that this is patient specific.

For now, some enhancements are needed to be usable in real life such as optimizing the prediction to be sure that the tumor is included in the ellipse. The time needed to predict and to refine the contour also needs to be quantified and optimized to be as short as possible. Moreover, our algorithm is not automatic yet since we would like that the data used in the predictor are based on results of the segmentation algorithm. The inverse is already the case, we use predictions to initialise segmentation algorithms.

Bibliography

[1] P.-A. Absil. Kalman filtering, 7 March 2020. Notes provided in the course LINMA1731 – Stochastic processes : Estimation and prediction.

[2] Damien Dasnoy-Sumell Anne-Sophie Collin. Challenge 5 : Active contours (snakes). In *LGBIO2050 - Medical imaging*. 2020-2021.

[3] Giuseppe Armenise, Marco Vaccari, Riccardo Bacci Di Capaci, and Gabriele Pannocchia. An open-source system identification package for multivariable processes. pages 152–157. IEEE, 9 2018.

[4] Antoine Aspeel, Benoît Macq, Raphaël Jungers, and Damien Dasnoy. Optimal intermittent measurements for tumor tracking in x-ray guided radiotherapy. page 11. SPIE, 3 2019.

[5] Venkataramanan Balakrishnan. System identification: theory for the user (second edition). *Automatica*, 38:375–378, 2 2002.

[6] Dan Bao, Zhou Liu, Yayuan Geng, Lin Li, Haijun Xu, Ya Zhang, Lei Hu, Xinming Zhao, Yanfeng Zhao, and Dehong Luo. Baseline mri-based radiomics model assisted predicting disease progression in nasopharyngeal carcinoma patients with complete response after treatment. *Cancer Imaging*, 22:10, 12 2022.

[7] J. Bellec, F. Arab-Ceschia, J. Castelli, C. Lafond, and E. Chajon. ITV versus mid-ventilation for treatment planning in lung SBRT: a comparison of target coverage and PTV adequacy by using in-treatment 4D cone beam CT. *Radiation Oncology*, 15(1):54, dec 2020.

[8] Jenny Bertholet, Antje Knopf, Björn Eiben, Jamie McClelland, Alexander Grimwood, Emma Harris, Martin Menten, Per Poulsen, Doan Trang Nguyen, Paul Keall, and Uwe Oelfke. Real-time intrafraction motion monitoring in external beam radiotherapy. *Physics in Medicine Biology*, 64:15TR01, 8 2019.

[9] Serge Beucher. The watershed transformation. `https://people.cmm.minesparis.psl.eu/users/beucher/wtshed.html`, May 2010.

[10] Lau Brix, Steffen Ringgaard, Thomas Sangild Sørensen, and Per Rugaard Poulsen. Three-dimensional liver motion tracking using real-time two-dimensional mri. *Medical Physics*, 41:042302, 3 2014.

[11] Laura I Cerviño, Jiang Du, and Steve B Jiang. Mri-guided tumor tracking in lung cancer radiotherapy. *Physics in Medicine and Biology*, 56:3773–3785, 7 2011.

[12] D. Dasnoy-Sumell, A. Aspeel, K. Souris, and B. Macq. Locally tuned deformation fields combination for 2d cine-mri-based driving of 3d motion models. *Physica Medica*, 94:8–16, 2 2022.

[13] Damien Dasnoy-Sumell. *Towards real-time image-guided radiotherapy : 3D patient-specific breathing motion models driven by 2D cine-MRI*. PhD thesis, UCL, 2021.

[14] Damien Dasnoy[U+2010]Sumell, Kevin Souris, G. Van Ooteghem, and Benoit Macq. Continuous real time 3d motion reproduction using dynamic mri and precomputed 4dct deformation fields. *Journal of Applied Clinical Medical Physics*, 21:236–248, 8 2020.

[15] Tom Eelbode, Jeroen Bertels, Maxim Berman, Dirk Vandermeulen, Frederik Maes, Raf Bisschops, and Matthew B. Blaschko. Optimization for medical image segmentation: Theory and practice when evaluating with dice score or jaccard index. *IEEE Transactions on Medical Imaging*, 39:3679–3690, 11 2020.

[16] Xiaojie Fan, Xiaoyu Zhang, Zibo Zhang, and Yifang Jiang. Deep learning on mri images for diagnosis of lung cancer spinal bone metastasis. *Contrast Media Molecular Imaging*, 2021:1–9, 7 2021.

[17] Mei Fang, GuangXue Yue, and QingCang Yu. The study on an application of otsu method in canny operator. 01 2009.

[18] Martin F. Fast, Björn Eiben, Martin J. Menten, Andreas Wetscherek, David J. Hawkes, Jamie R. McClelland, and Uwe Oelfke. Tumour auto-contouring on 2d cine MRI for locally advanced lung cancer: A comparative study. *Radiotherapy and Oncology*, 125(3):485–491, dec 2017.

[19] William Franklin. The kalman filter : a beginners guide to the kalman filter. `https://thekalmanfilter.com/kalman-filter-python-example/`, 2021.

[20] Nathan Funk. A study of the kalman filter applied to visual tracking. 01 2003.

[21] Michael Galarnyk. Understanding boxplots. `https://towardsdatascience.com/understanding-boxplots-5e2df7bcbd51`, September 2018.

[22] Yuanyuan Ge, Ricky T. O'Brien, Chun-Chien Shieh, Jeremy T. Booth, and Paul J. Keall. Toward the development of intrafraction tumor deformation tracking using a dynamic multi-leaf collimator. *Medical Physics*, 41:061703, 5 2014.

[23] Lei He, Zhigang Peng, Bryan Everding, Xun Wang, Chia Y. Han, Kenneth L. Weiss, and William G. Wee. A comparative study of deformable contour methods on medical image segmentation. *Image and Vision Computing*, 26(2):141–163, 2008.

[24] Hanne D. Heerkens, Marco van Vulpen, Cornelis A.T. van den Berg, Rob H.N. Tijssen, Sjoerd P.M. Crijns, Izaak Q. Molenaar, Hjalmar C. van Santvoort, Onne Reerink, and Gert J. Meijer. Mri-based tumor motion characterization and gating schemes for radiation therapy of pancreatic cancer. *Radiotherapy and Oncology*, 111:252–257, 5 2014.

[25] Peter Hunold, Stefan Maderwald, Mark E. Ladd, Vladimir Jellus, and J[U+FFFD]rg Barkhausen. Parallel acquisition techniques in cardiac cine magnetic resonance imaging using TrueFISP sequences: Comparison of image quality and artifacts. *Journal of Magnetic Resonance Imaging*, 20(3):506–511, sep 2004.

[26] IAEA. *Radiotherapy in Cancer Care: Facing the Global Challenge*. ProQuest Ebook Central, 2017.

[27] Julius O. Smith III. Markov parameters. In *Introduction to Digital Filters with Audio Applications*. Center for Computer Research in Music and Acoustics (CCRMA), Stanford University, 2007.

[28] Jue Jiang, Yu[U+2010]Chi Hu, Neelam Tyagi, Pengpeng Zhang, Andreas Rimner, Joseph O. Deasy, and Harini Veeraraghavan. Cross[U+2010]modality (CT[U+2010]MRI) prior augmented deep learning for robust lung tumor segmentation from small MR datasets. *Medical Physics*, 46(10):4392–4404, oct 2019.

[29] A. Jouin and N. Pourel. Marges dans le cancer pulmonaire : volume cible interne/volume cible anatomoclinique. *Cancer/Radiothérapie*, 17(5-6):428–433, oct 2013.

[30] Bo-Hwan Jung, B.-H Kim, and S.-M Hong. Respiratory motion prediction with extended kalman filters based on local circular motion model. *International Journal of Bio-Science and Bio-Technology*, 5, 01 2013.

[31] Lukas Knybel, Jakub Cvek, Lukas Molenda, Natalie Stieberova, and David Feltl. Analysis of lung tumor motion in a large sample: Patterns and factors influencing precise delineation of internal target volume. *International Journal of Radiation Oncology*Biology*Physics*, 96:751–758, 11 2016.

[32] W.E. Larimore. Canonical variate analysis in identification, filtering, and adaptive control. In *29th IEEE Conference on Decision and Control*, pages 596–604 vol.2, 1990.

[33] Verleysen Michel Lee John. Principal component analysis (pca). In *LELEC2870 Machine learning : regression, deep networks and dimensionality reduction*. UCLouvain, November 2021.

[34] Qiang Li, Ranyang Li, Kaifan Ji, and Wei Dai. Kalman filter and its application. pages 74–77. IEEE, 11 2015.

[35] Benoit Lombardot. Interactive watershed. `https://imagej.net/plugins/interactive-watershed`, January 2018.

[36] Zhen Ma, João Manuel R.S. Tavares, Renato Natal Jorge, and T. Mascarenhas. A review of algorithms for medical image segmentation and their applications to the female pelvic cavity. *Computer Methods in Biomechanics and Biomedical Engineering*, 13(2):235–246, apr 2010.

[37] Abolfazl Meyarian, Himan Namdari, Xiaohui Yuan, Mark V. Albert, and John C. Roeske. Phantom tumor tracking in dual-energy fluoroscopy using a kalman filter. pages 2023–2026. IEEE, 12 2020.

[38] Gautam S. Muralidhar, Tejaswini Ganapathi, Alan C. Bovik, Mia K. Markey, Tamara M. Haygood, Tanya W. Stephens, and Gary J. Whitman. Stereoscopic versus monoscopic detection of masses on breast tomosynthesis projection images. page 831807, 2 2012.

[39] Peter Van Overschee and Bart De Moor. N4sid: Subspace algorithms for the identification of combined deterministic-stochastic systems. *Automatica*, 30:75–93, 1 1994.

[40] Gabriele Pannocchia and Mirco Calosi. A predictor form PARSIMonious algorithm for closed-loop subspace identification. *Journal of Process Control*, 20(4):517–524, apr 2010.

[41] S. Joe Qin. An overview of subspace identification. *Computers Chemical Engineering*, 30:1502–1513, 9 2006.

[42] Giuseppe Armenise Gabriele Pannocchia Riccardo Bacci di Capaci, Marco Vaccari. *Systems Identification Package for PYthon (SIPPY) : user guide*, April 2021.

[43] Lennart Ljung S. Joe Qin. Identification with parsimonious models. Technical report, Department of Electrical Engineering, Linköpings university, Linköping, Sweden, December 2003.

[44] Lennart Ljung S. Joe Qin, Weilu Lin. A novel subspace identification approach with enforced causal models. Technical report, Department of Electrical Engineering, Linköpings university, Linköping, Sweden, June 2007.

[45] Gregory C Sharp, Steve B Jiang, Shinichi Shimizu, and Hiroki Shirato. Prediction of respiratory tumour motion for real-time image-guided radiotherapy. *Physics in Medicine and Biology*, 49:425–440, 2 2004.

[46] Shinichi Shimizu, Hiroki Shirato, Shigeaki Ogura, Hirotoshi Akita-Dosaka, Kei Kitamura, Takeshi Nishioka, Kenji Kagei, Masaji Nishimura, and Kazuo Miyasaka. Detection of lung tumor movement in real-time tumor-tracking radiotherapy. *International Journal of Radiation Oncology*Biology*Physics*, 51:304–310, 10 2001.

[47] Steven H. Strogatz. *Nonlinear dynamics and chaos with Applications to Physics, Biology, Chemistry, and Engineering*. CRC Press, 2015.

[48] Nazanin Tahmasebi, Pierre Boulanger, Jihyun Yun, Gino Fallone, Michelle Noga, and Kumaradevan Punithakumar. Real-time lung tumor tracking using a cuda enabled nonrigid registration algorithm for mri. *IEEE Journal of Translational Engineering in Health and Medicine*, 8:1–8, 2020.

[49] Geneviève Van Ooteghem, Damien Dasnoy-Sumell, Maarten Lambrecht, Grégory Reychler, Giuseppe Liistro, Edmond Sterpin, and Xavier Geets. Mechanically-assisted non-invasive ventilation: A step forward to modulate and to improve the reproducibility of breathing-related motion in radiation therapy. *Radiotherapy and Oncology*, 133:132–139, apr 2019.

[50] S. S. Vedam, P. J. Keall, A. Docef, D. A. Todor, V. R. Kini, and R. Mohan. Predicting respiratory motion for four-dimensional radiotherapy. *Medical Physics*, 31(8):2274–2283, jul 2004.

[51] Michel Verhaegen and Patrick Dewilde. Subspace model identification part 1. the output-error state-space model identification class of algorithms. *International Journal of Control*, 56:1187–1210, 11 1992.

[52] Aryan Verma. Implement canny edge detector in python using opencv. `https://www.geeksforgeeks.org/implement-canny-edge-detector-in-python-using-opencv/`, 2022.

[53] Donna J. Williams and Mubarak Shah. A fast algorithm for active contours and curvature estimation. *CVGIP: Image Understanding*, 55:14–26, 1 1992.

[54] Andrew Witkin and Demetri Terzopoulo Michael Kass. Snakes: Active contour models. *International Journal of Computer Vision*, pages 321–331, 1988.

[55] Dehui Xiang, Bolan Yang, Feihong Yu, and Xinjian Chen. Lung tumor segmentation based on multi-scale template matching and region growing. In Barjor Gimi and Andrzej Krol, editors, *Medical Imaging 2018: Biomedical Applications in Molecular, Structural, and Functional Imaging*, page 98. SPIE, mar 2018.

[56] Jihyun Yun, Eugene Yip, Keith Wachowicz, Satyapal Rathee, Marc Mackenzie, Don Robinson, and B. G. Fallone. Evaluation of a lung tumor autocontouring algorithm for intrafractional tumor tracking using low-field MRI: A phantom study. *Medical Physics*, 39(3):1481–1494, feb 2012.

Appendix A

Trajectories

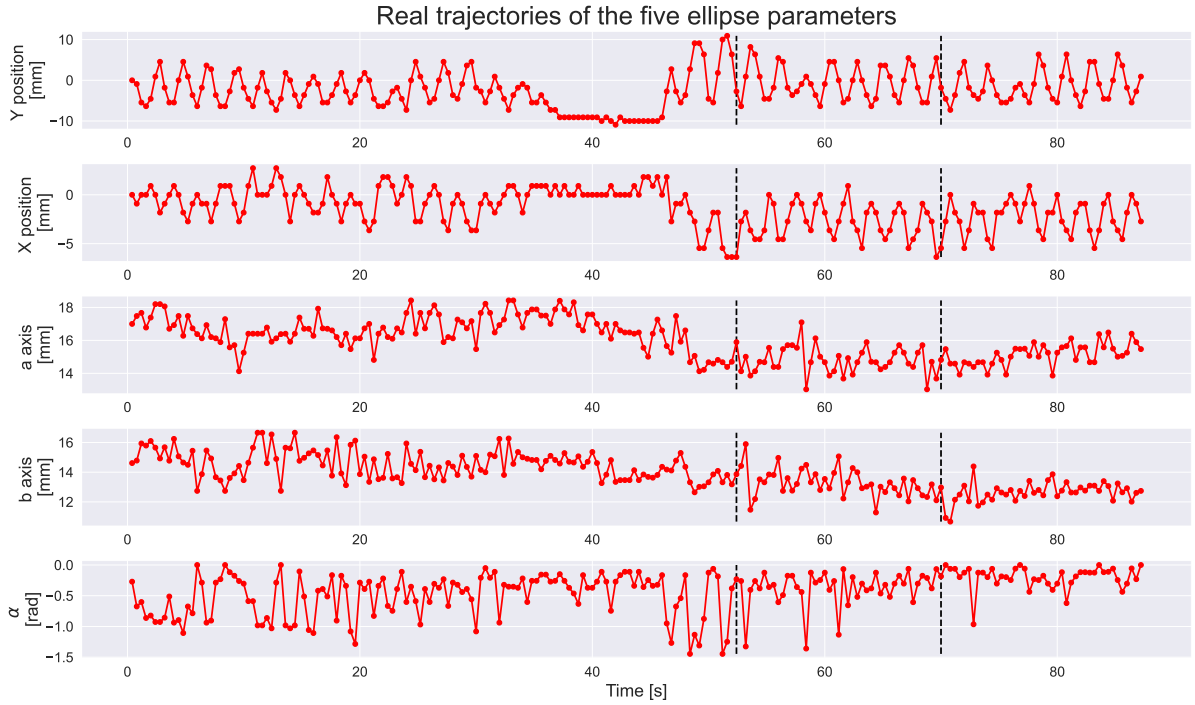

Figure A.1: Trajectories for the five parameters of the ellipse. The vertical lines represent the separation between the training, validation and testing sets. P1S. All sets.

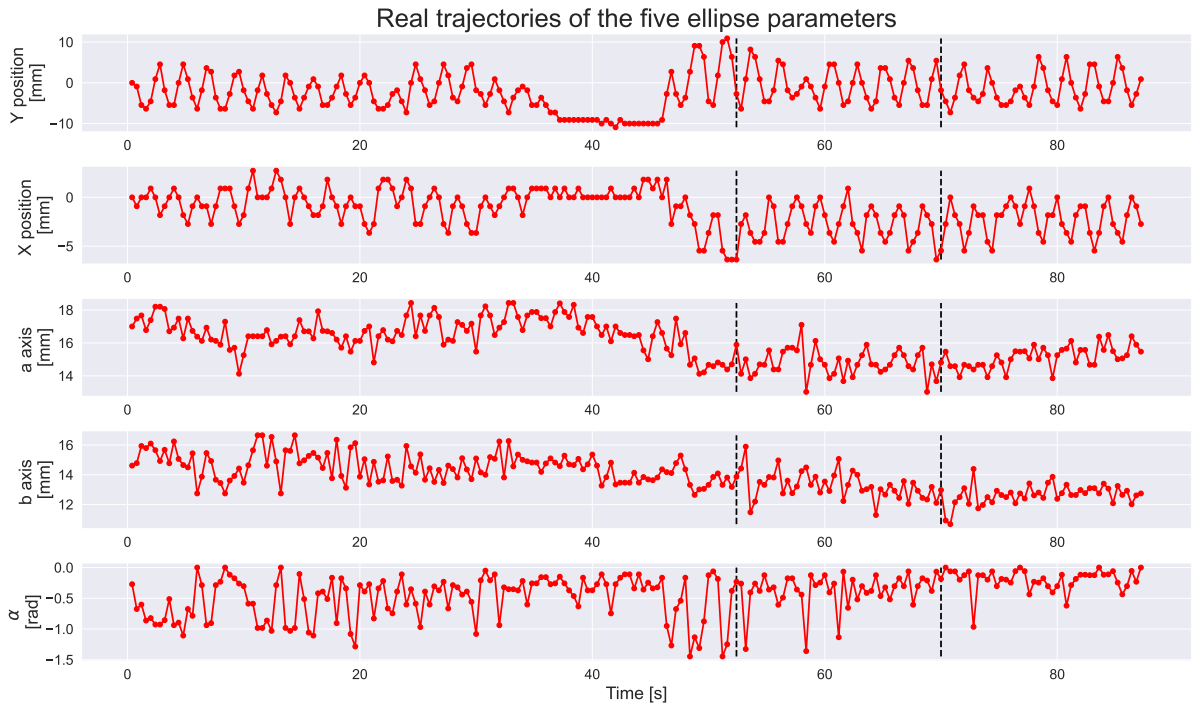

Figure A.2: Trajectories for the five parameters of the ellipse. The vertical lines represent the separation between the training, validation and testing sets. P2S1. All sets.

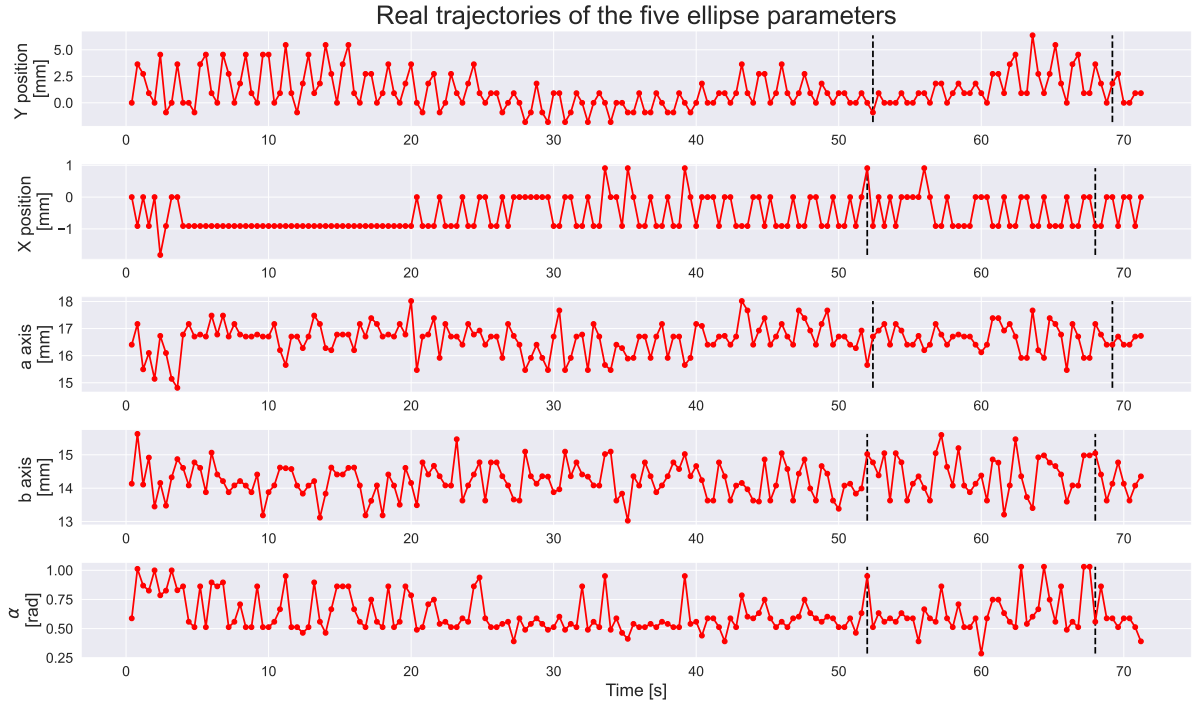

Figure A.3: Trajectories for the five parameters of the ellipse. The vertical lines represent the separation between the training, validation and testing sets. P2S2. All sets.

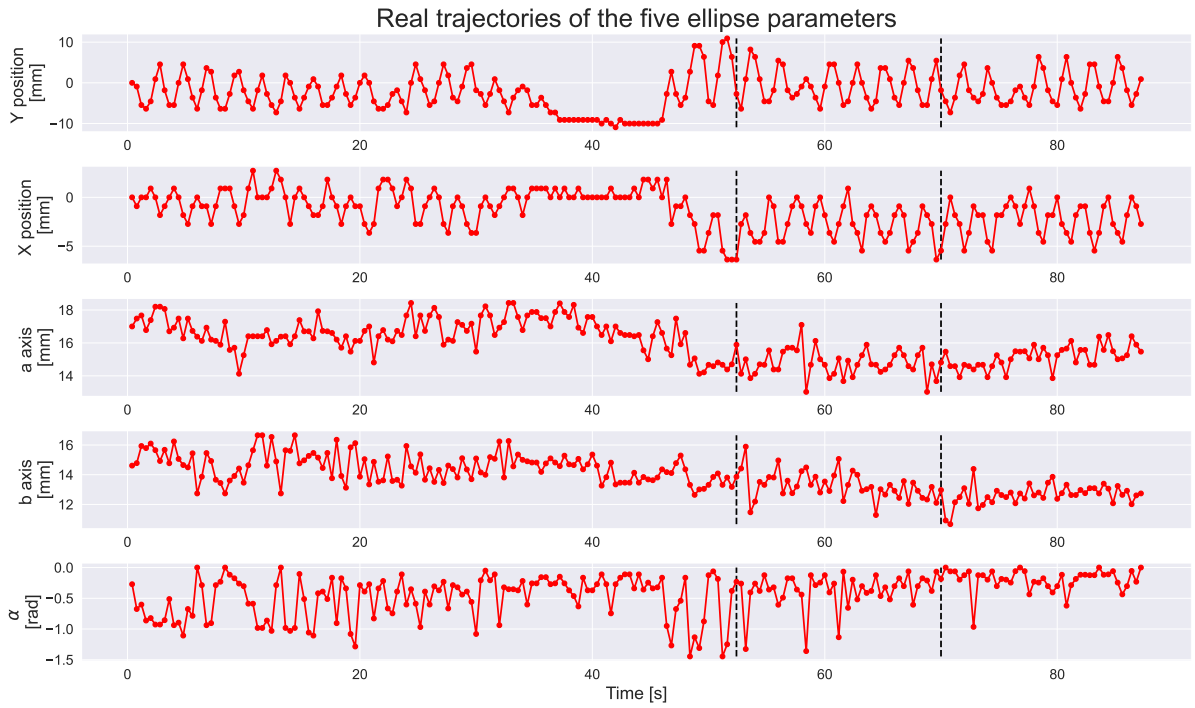

Figure A.4: Trajectories for the five parameters of the ellipse. The vertical lines represent the separation between the training, validation and testing sets. P2S3. All sets.

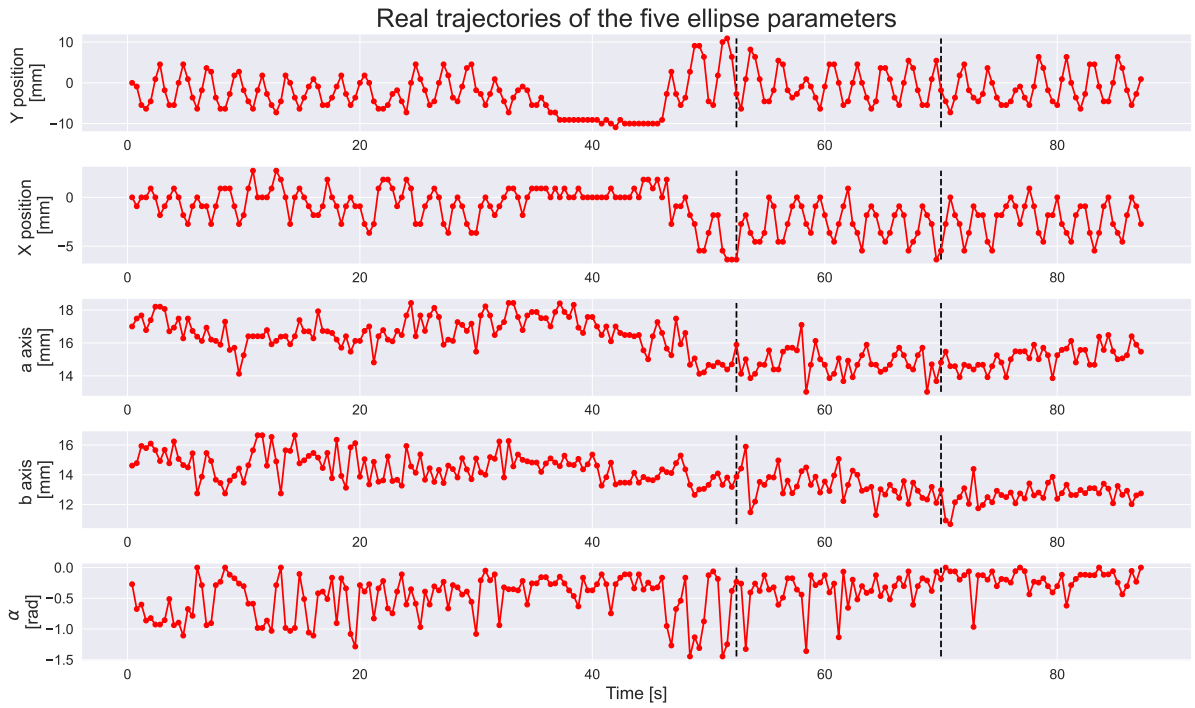

Figure A.5: Trajectories for the five parameters of the ellipse. The vertical lines represent the separation between the training, validation and testing sets. P3C. All sets.

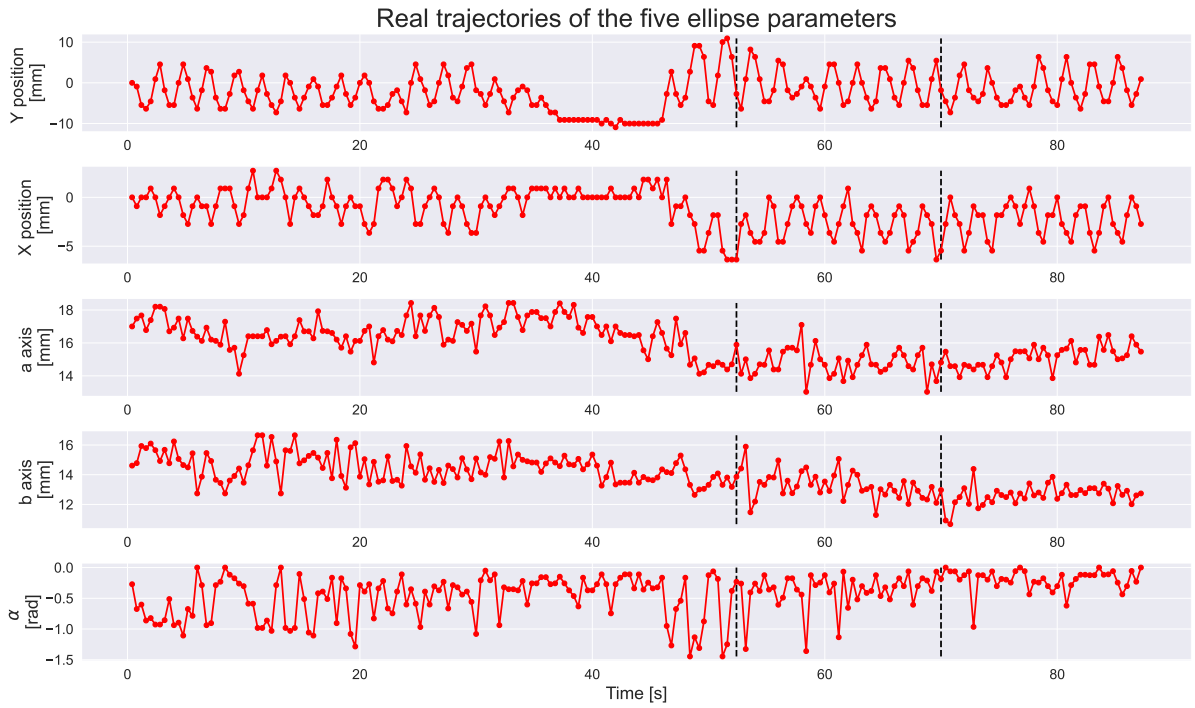

Figure A.6: Trajectories for the five parameters of the ellipse. The vertical lines represent the separation between the training, validation and testing sets. P3S. All sets.

Appendix B

Sinus model

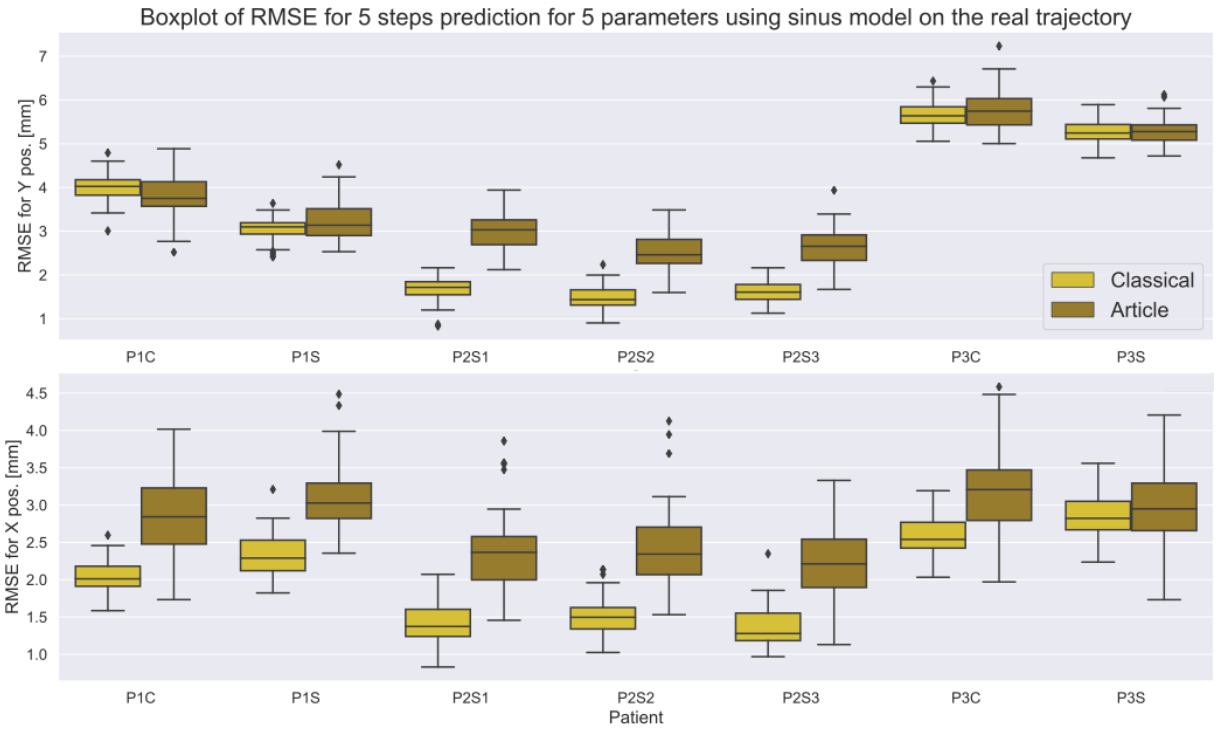

Figure B.1: Boxplot of RMSE for 5 steps prediction for 5 parameters using sinus models on the real trajectory. All patients. Evaluation set. 50 simulations.

Appendix C

Kalman filter: Analytical

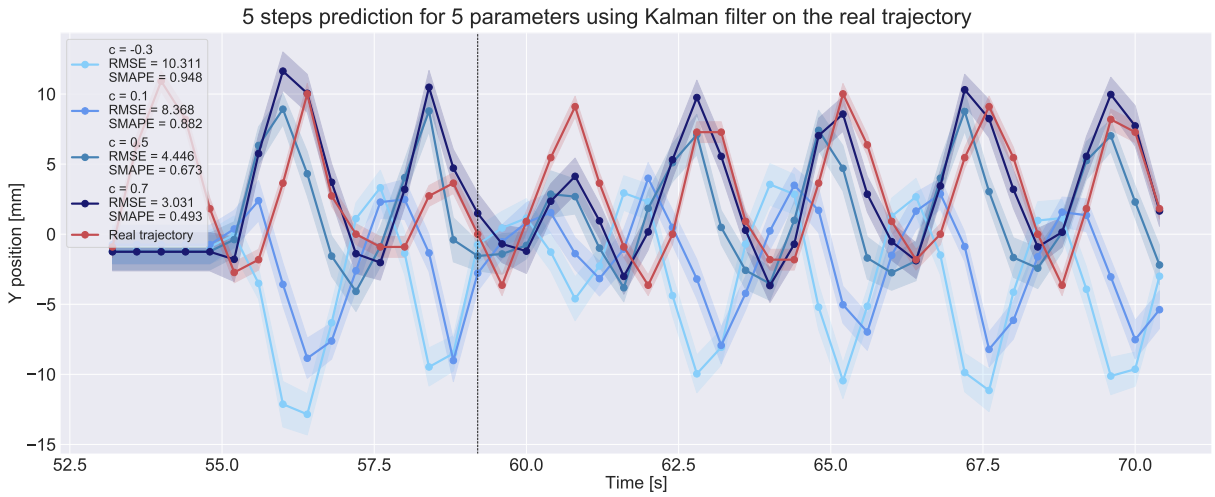

Figure C.1: 5 steps prediction for 5 parameters using Kalman filter on the real trajectory. The model is found analytically. Three values of parameter c are used: -0.3, 0.1, 0.5 and 0.7. P3C. Testing set. 100 simulations.

Appendix D

Kalman filter: System identification

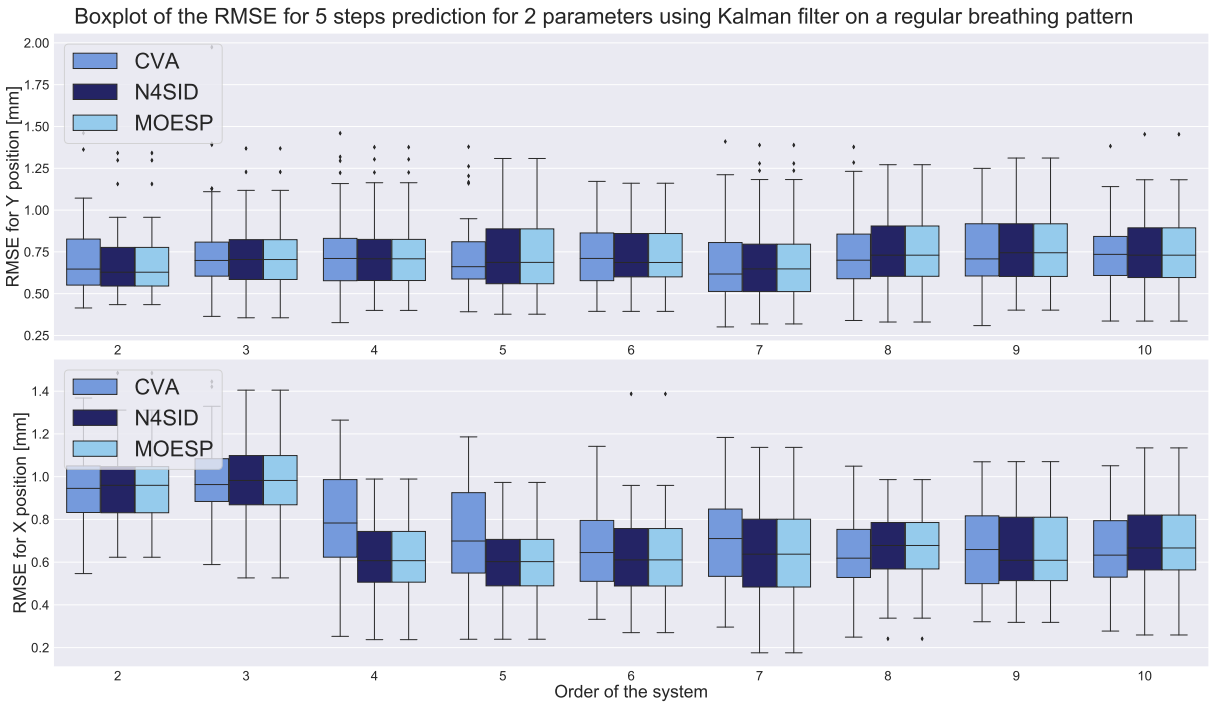

Figure D.1: Boxplot of the RMSE for 5 steps prediction for 2 parameters using Kalman filter on a regular breathing pattern. The model is found with a system identification and three identification methods are used. Left: MOESP/N4SID, Right: CVA. The order of the system varies from 2 to 10. P1C. Evaluation set. 100 simulations.

Appendix E

Comparison between patients

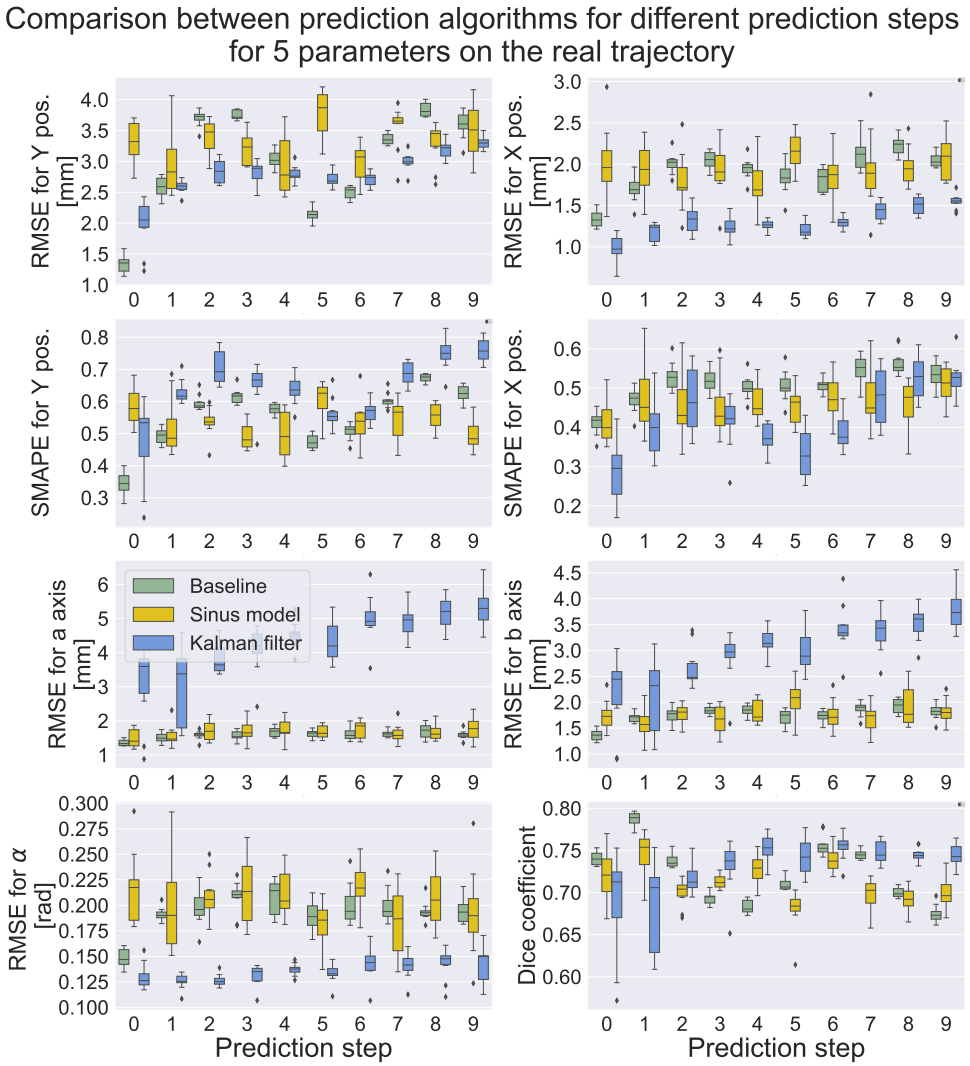

Figure E.1: Comparison between prediction algorithms for 5 parameters on the real trajectory. The prediction step varies from 0 to 9. P1S. Testing set. 100 simulations.

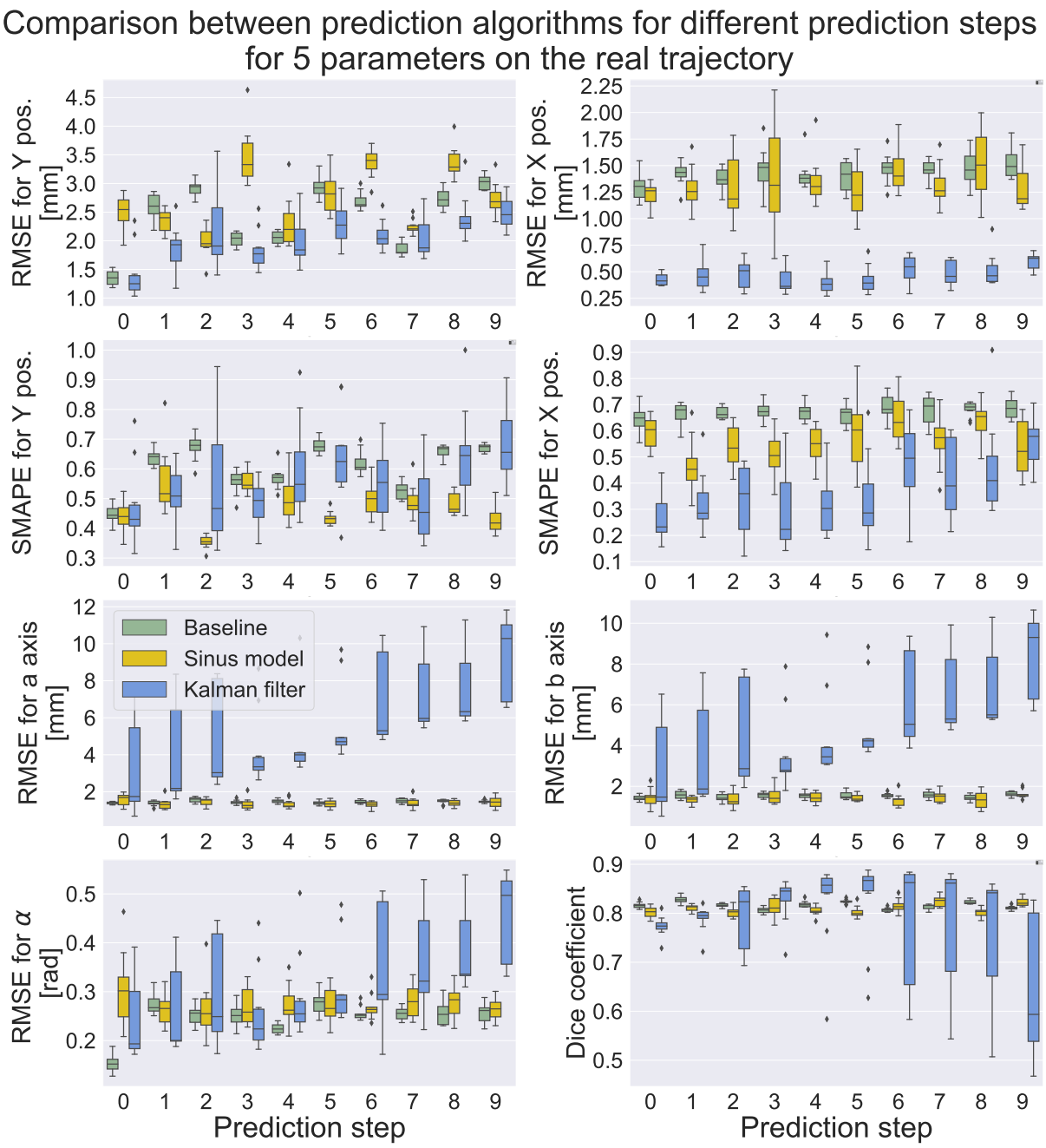

Figure E.2: Comparison between prediction algorithms for 5 parameters on the real trajectory. The prediction step varies from 0 to 9. P2S1. Testing set. 100 simulations.

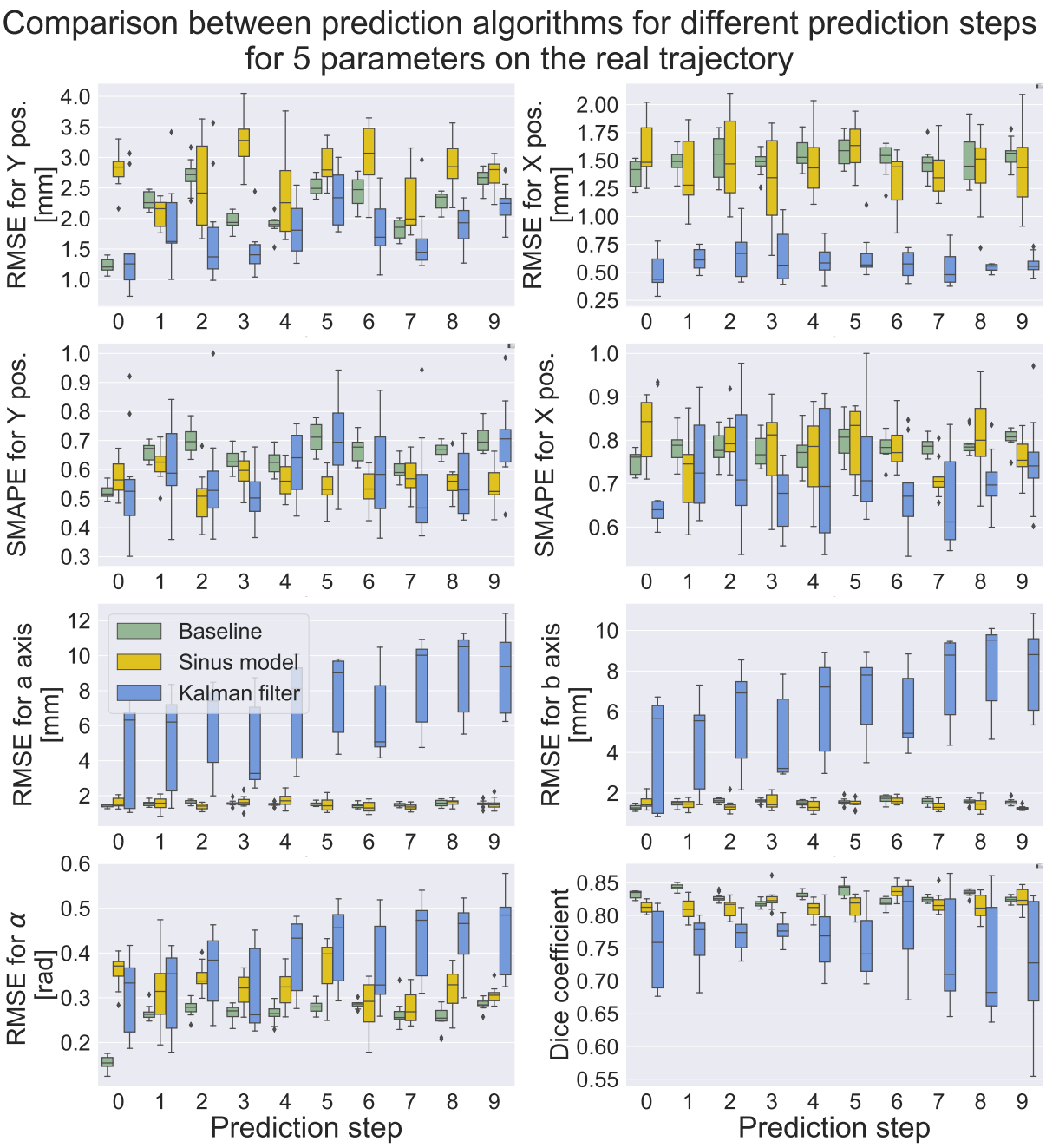

Figure E.3: Comparison between prediction algorithms for 5 parameters on the real trajectory. The prediction step varies from 0 to 9. P2S2. Testing set. 100 simulations.

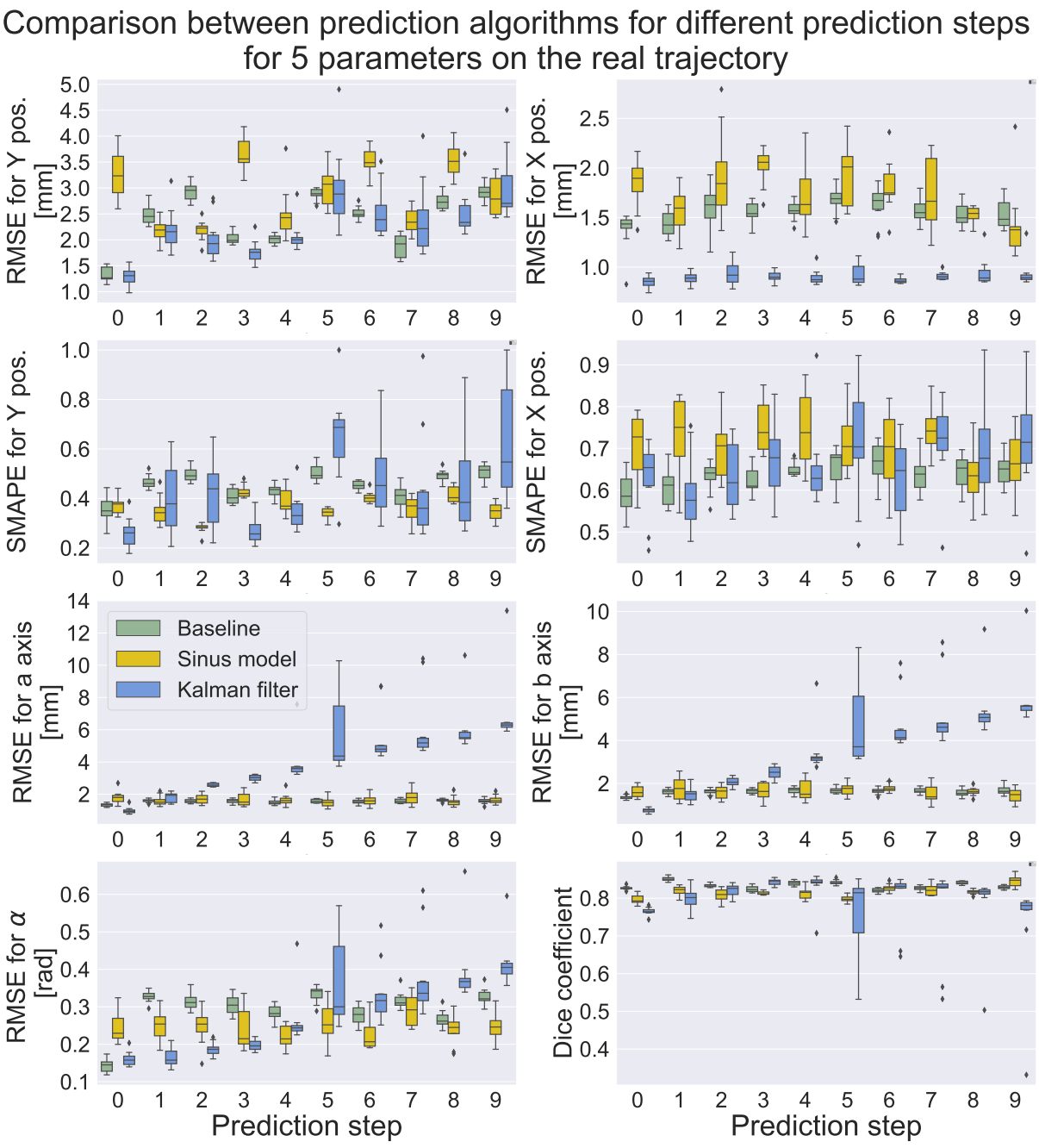

Figure E.4: Comparison between prediction algorithms for 5 parameters on the real trajectory. The prediction step varies from 0 to 9. P2S3. Testing set. 100 simulations.

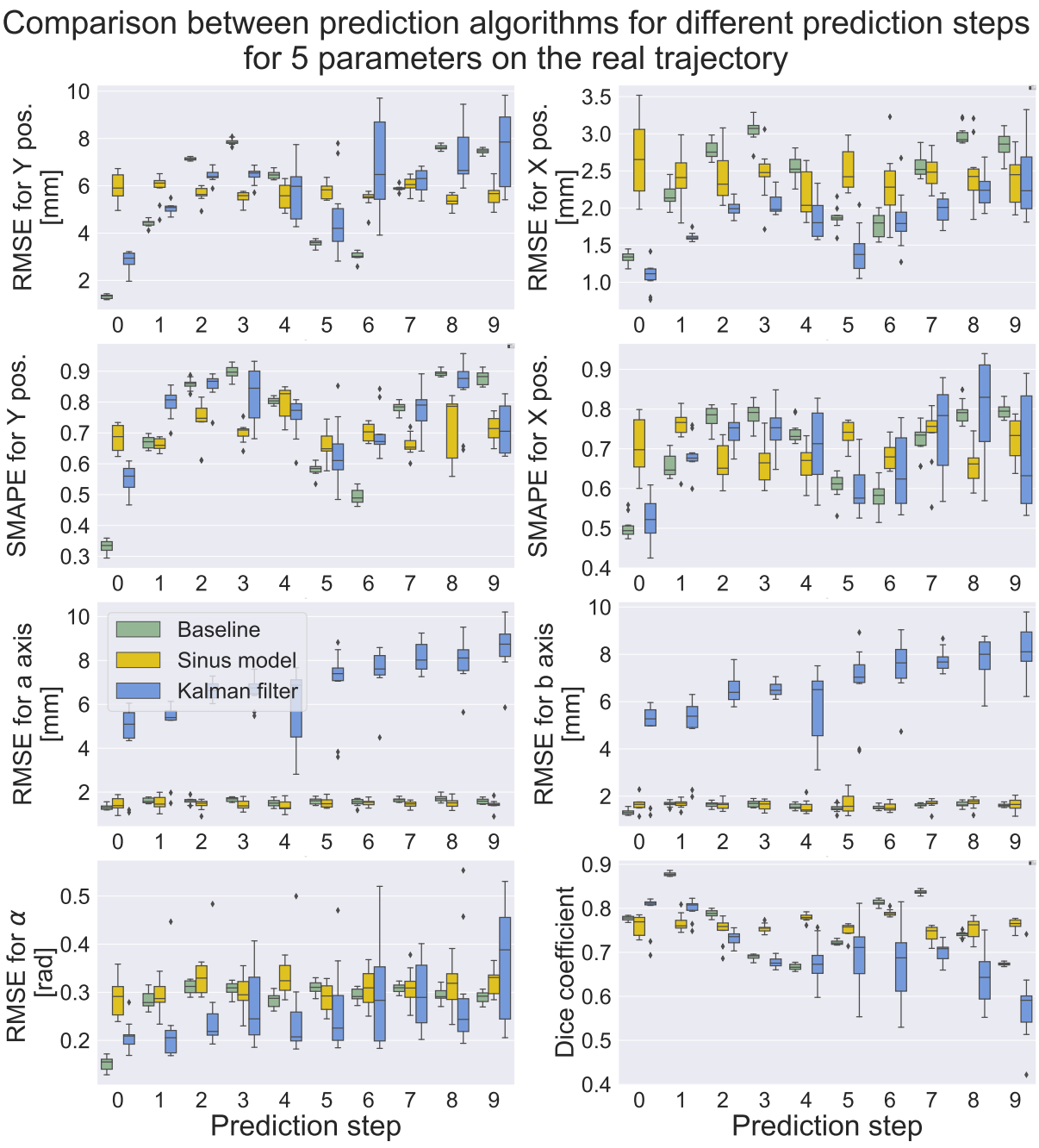

Figure E.5: Comparison between prediction algorithms for 5 parameters on the real trajectory. The prediction step varies from 0 to 9. P3C. Testing set. 100 simulations.

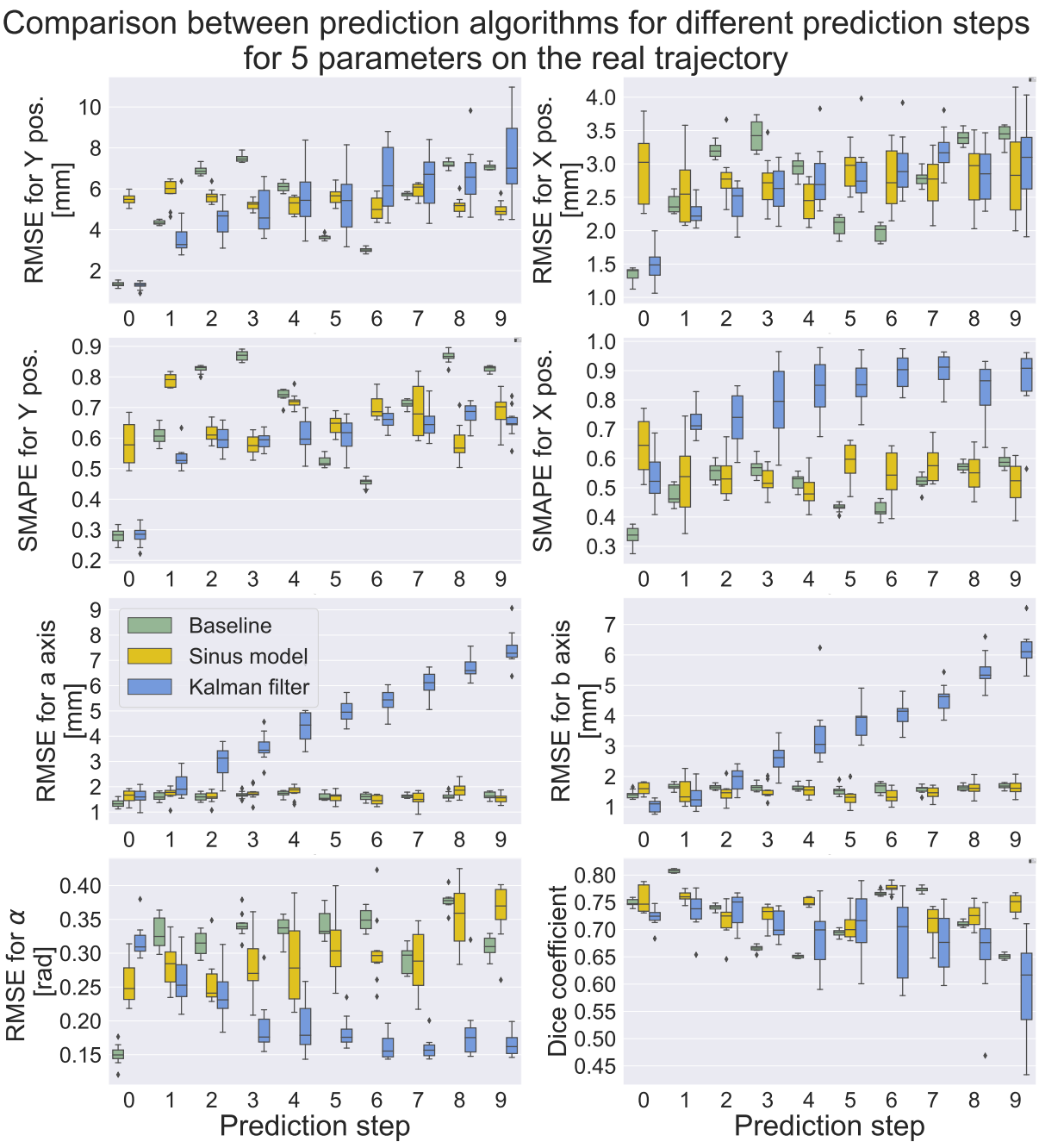

Figure E.6: Comparison between prediction algorithms for 5 parameters on the real trajectory. The prediction step varies from 0 to 9. P3S. Testing set. 100 simulations.

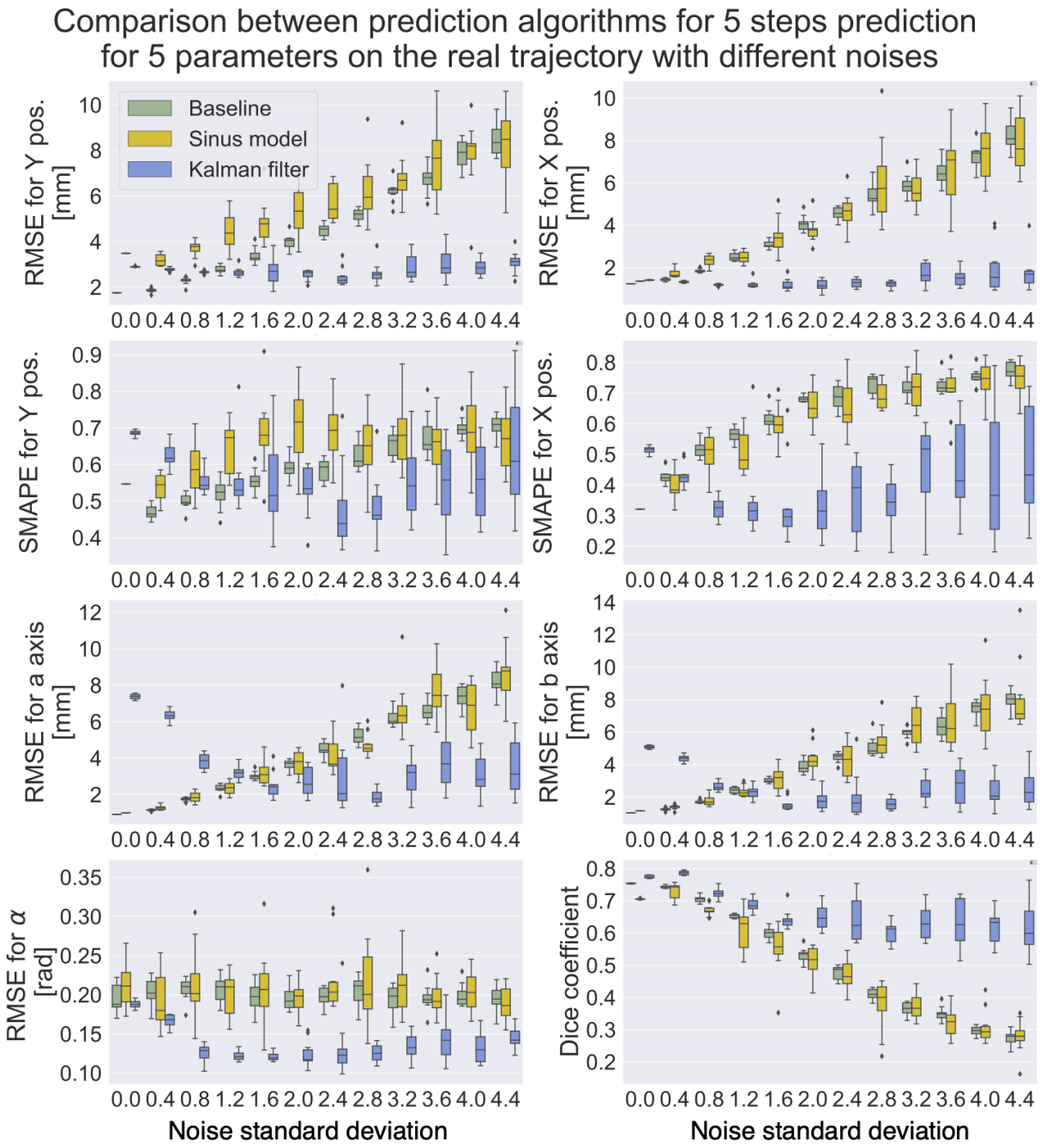

Figure E.7: Comparison between prediction algorithms for 5 parameters on the real trajectory. The noise standard deviation goes from 0 to 4.4. P1S. Testing set. 100 simulations.

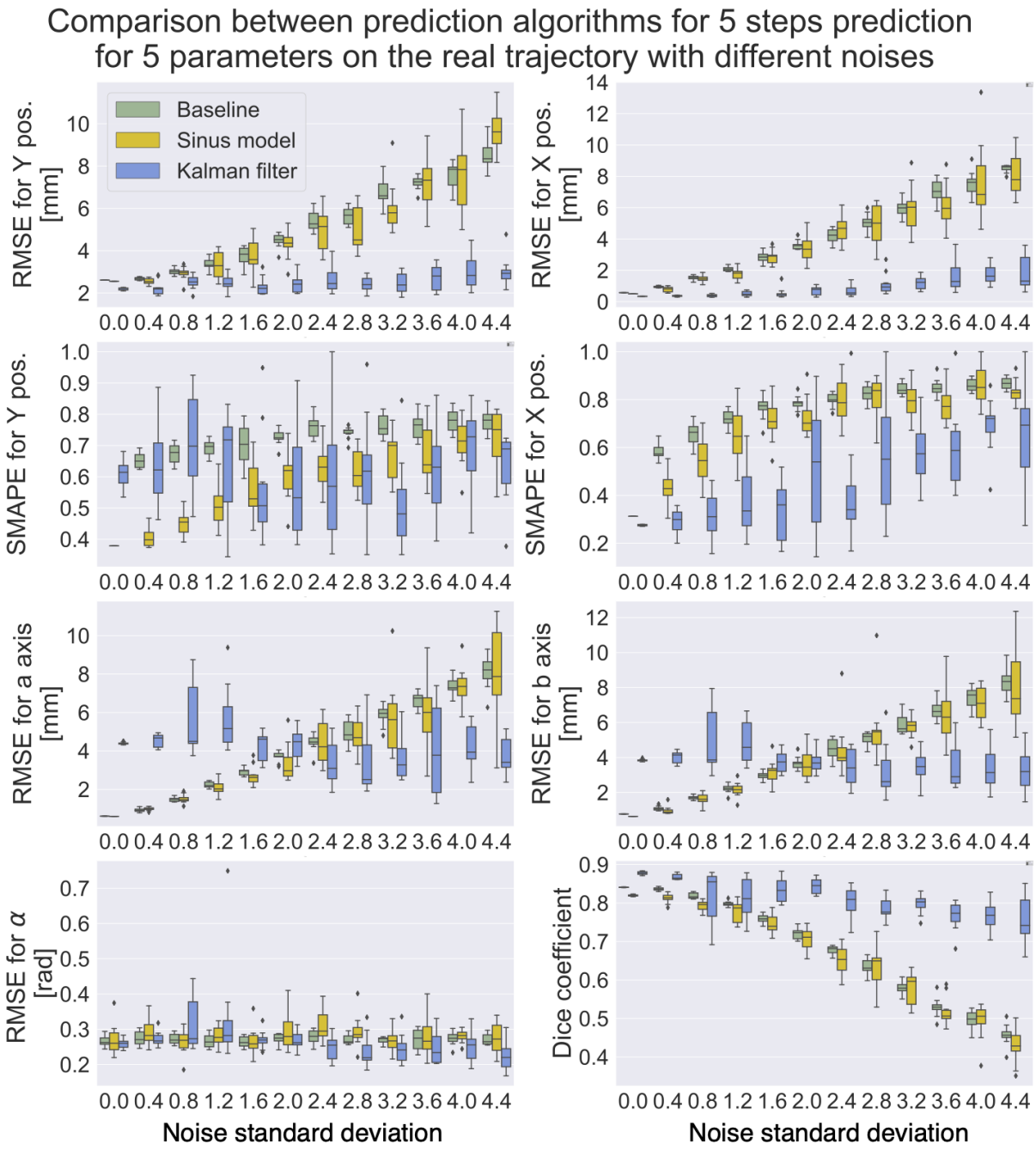

Figure E.8: Comparison between prediction algorithms for 5 parameters on the real trajectory. The noise standard deviation goes from 0 to 4.4. P2S1. Testing set. 100 simulations.

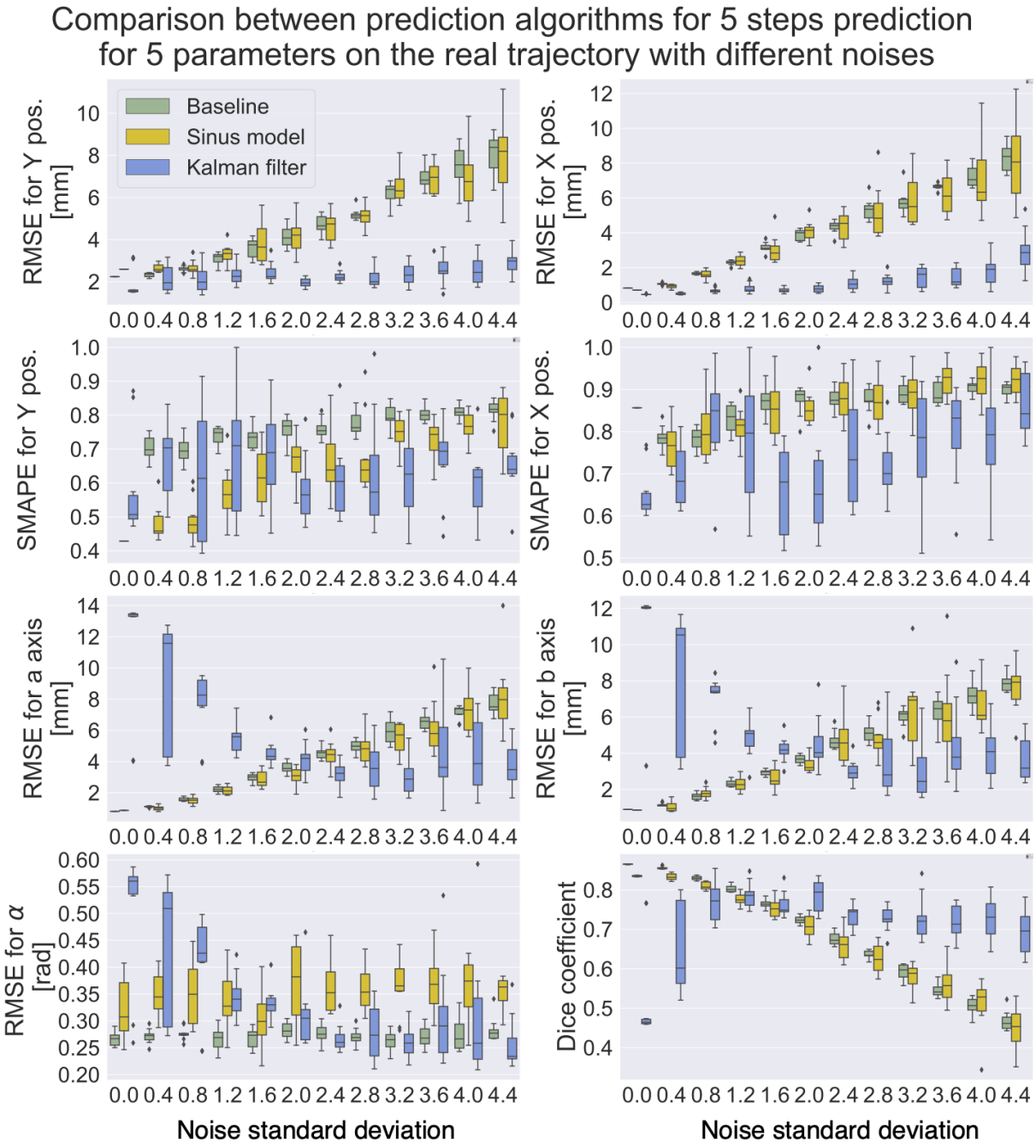

Figure E.9: Comparison between prediction algorithms for 5 parameters on the real trajectory. The noise standard deviation goes from 0 to 4.4. P2S2. Testing set. 100 simulations.

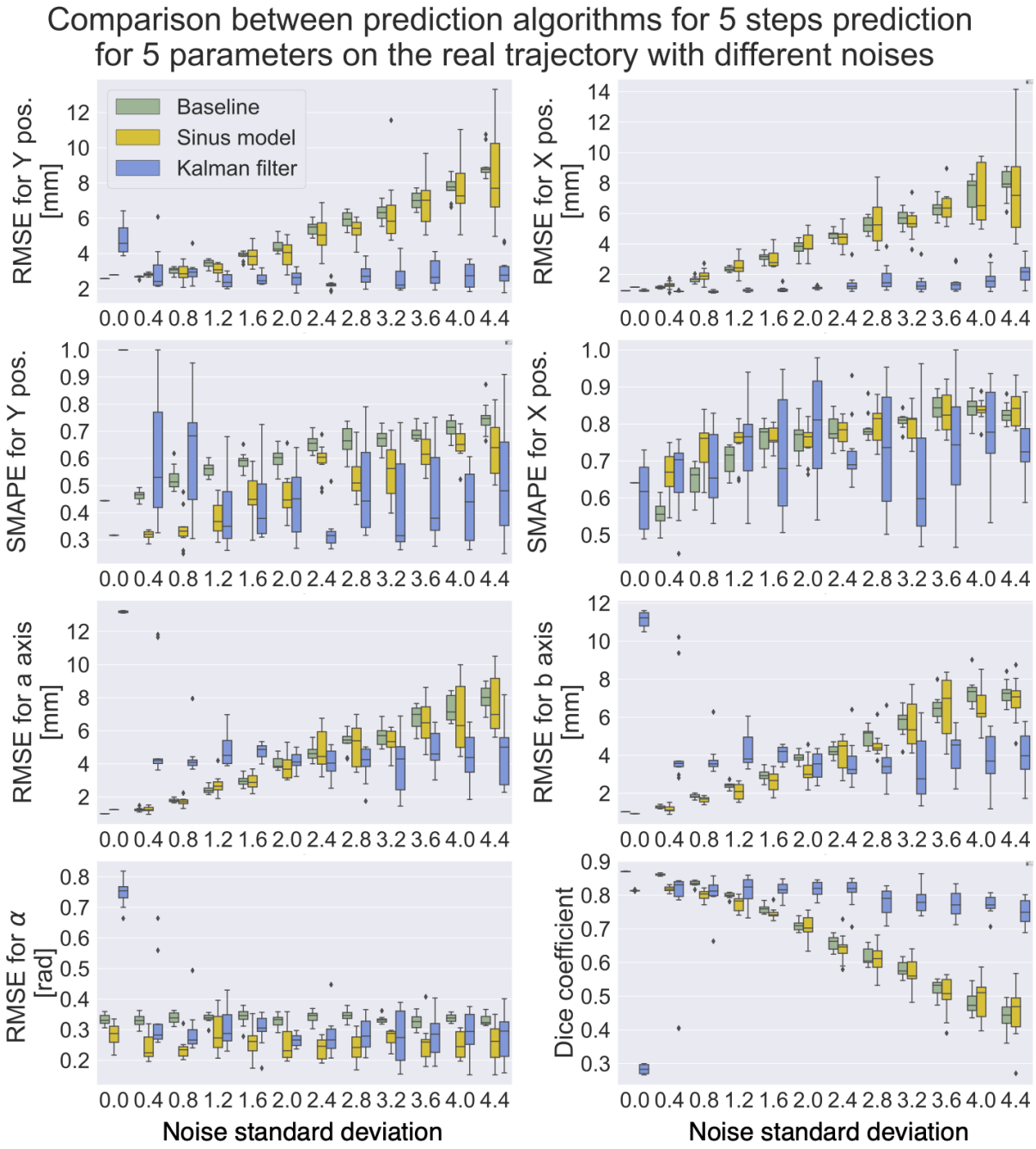

Figure E.10: Comparison between prediction algorithms for 5 parameters on the real trajectory. The noise standard deviation goes from 0 to 4.4. P2S3. Testing set. 100 simulations.

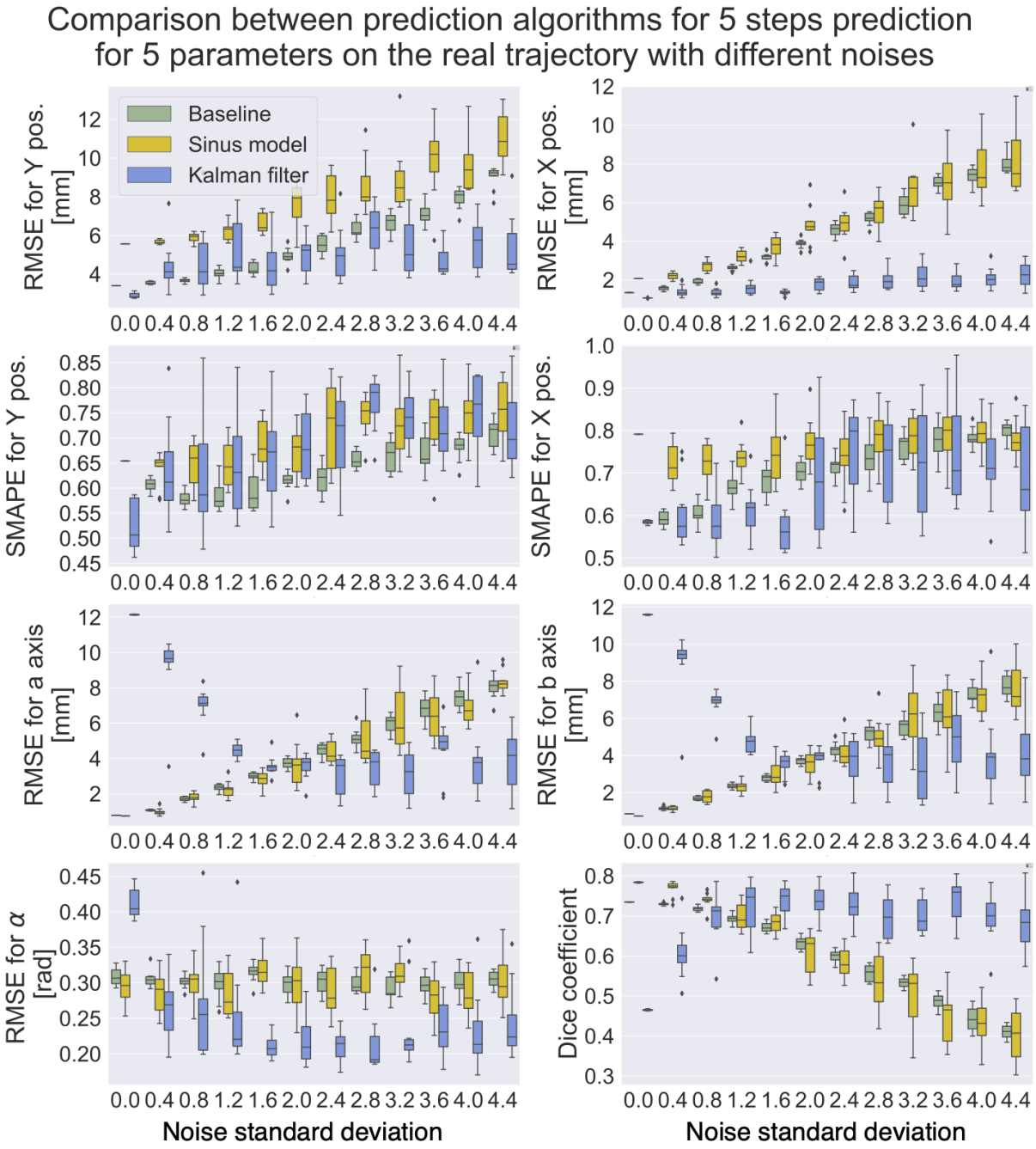

Figure E.11: Comparison between prediction algorithms for 5 parameters on the real trajectory. The noise standard deviation goes from 0 to 4.4. P3C. Testing set. 100 simulations.

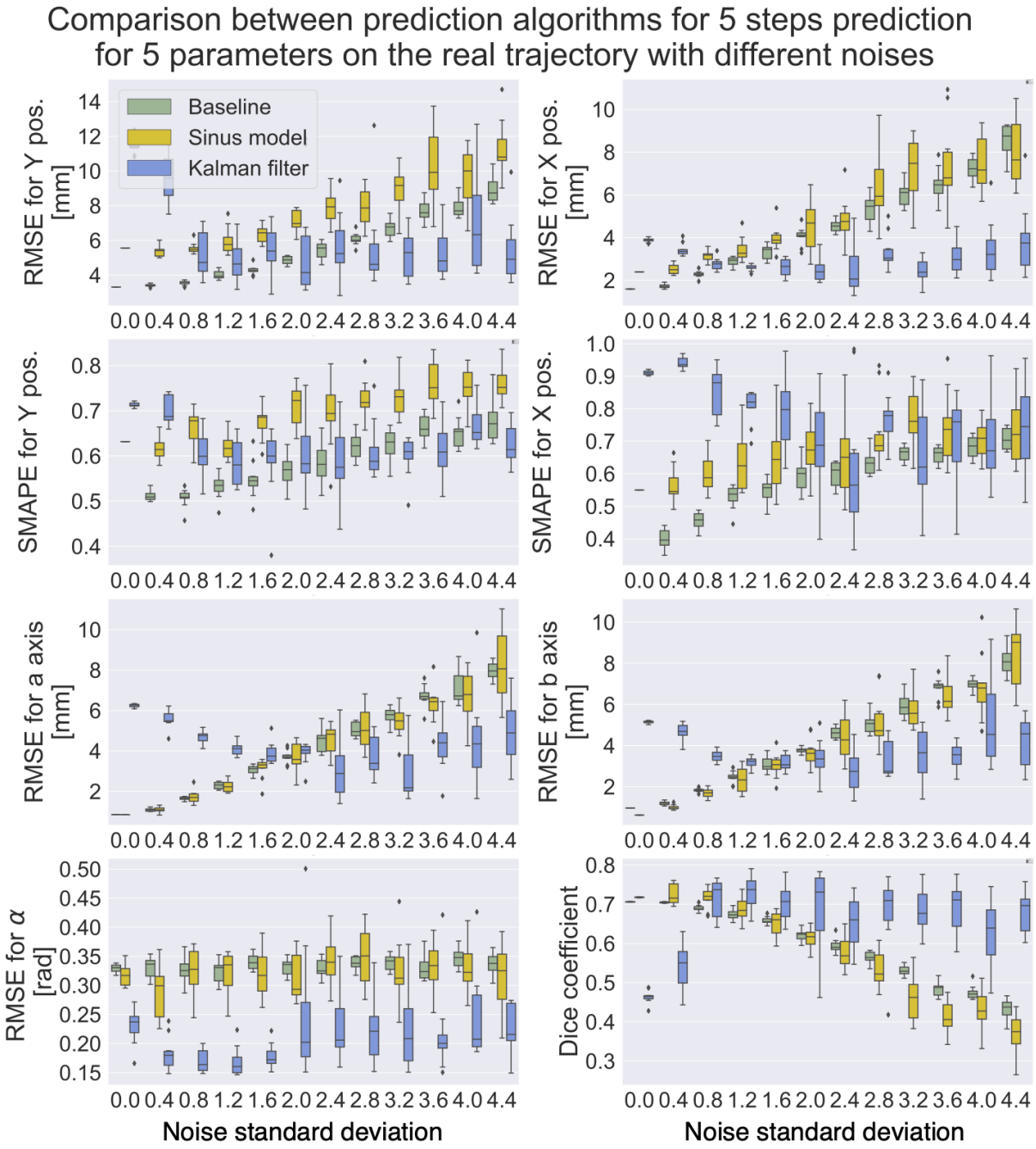

Figure E.12: Comparison between prediction algorithms for 5 parameters on the real trajectory. The noise standard deviation goes from 0 to 4.4. P3S. Testing set. 100 simulations.

Appendix F

Snake algorithm

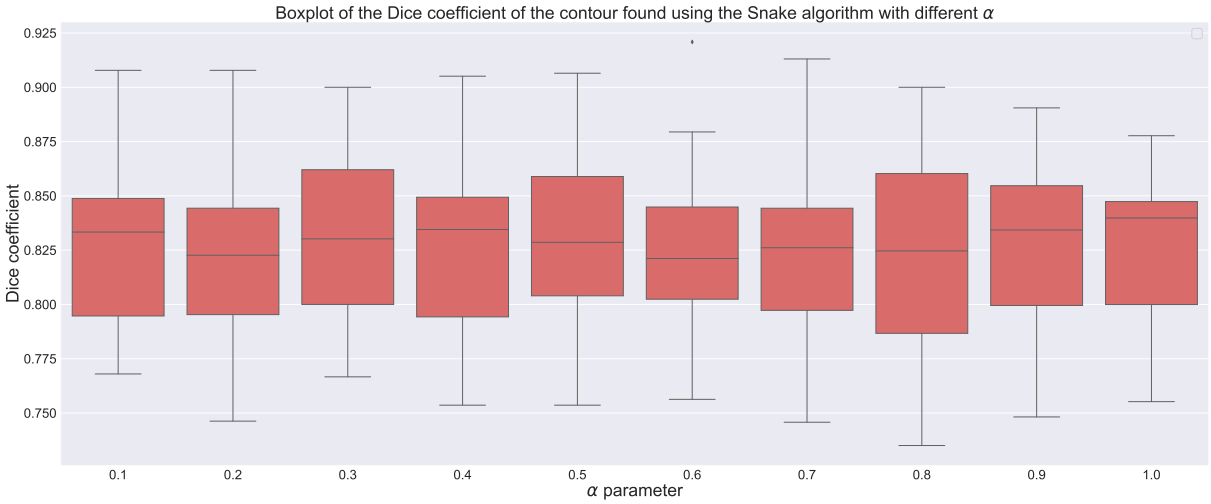

Figure F.1: Boxplot of the Dice coefficient of the contour found using the Snake algorithm with different α. The parameter α varies between 0.1 and 1.0; $\beta = 0.5$ and $\gamma = 0.5$. P1C. Training set.

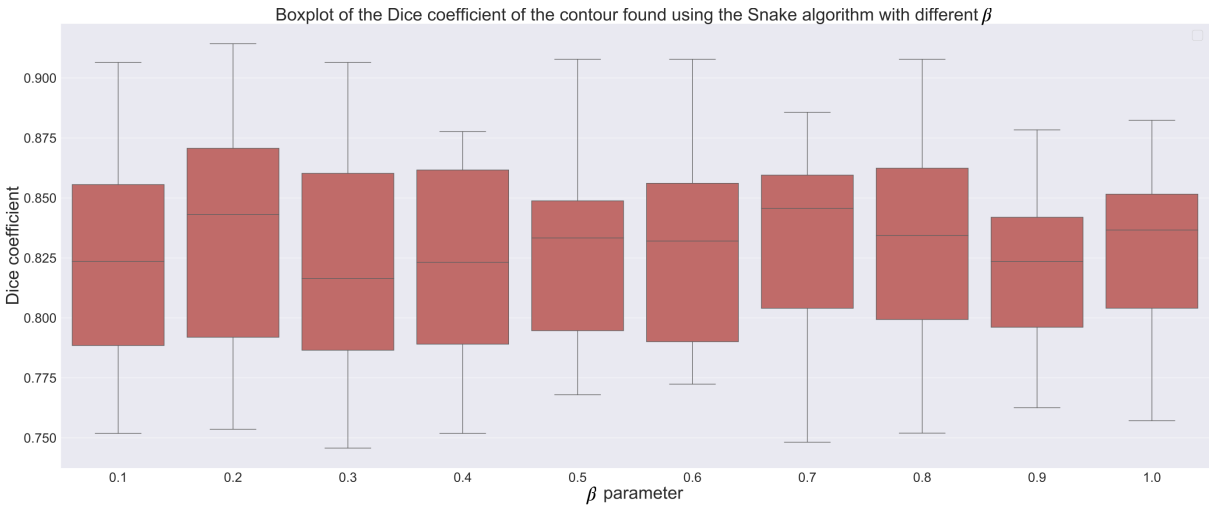

Figure F.2: Boxplot of the Dice coefficient of the contour found using the Snake algorithm with different β. The parameter β varies between 0.1 and 1.0; α is put at its optimal value, $\gamma = 0.5$. P1C. Training set.

UNIVERSITÉ CATHOLIQUE DE LOUVAIN
École polytechnique de Louvain
Rue Archimède, 1 bte L6.11.01, 1348 Louvain-la-Neuve, Belgique | **www.uclouvain.be/epl**