Óbuda University

PhD Thesis

Model-based control of cancerous diseases
by
Bence Géza Czakó

Supervisors:
Dr. Dániel András Drexler
Prof. Dr. Levente Kovács

Applied Informatics and Applied Mathematics Doctoral School
Budapest, 2023

Contents

List of figures

List of abbreviations

DSS Direct Single Shooting

DMS Direct Multiple Shooting

EM Expectation Maximization

FIE Full Information Estimator

IO Input-Output

KM Kaplan-Meier

MHE Moving Horizon Estimation

ML Maximum Likelihood

MTD Maximum Tolerable Dose

NLS Nonlinear Least-Squares

NMPC Nonlinear Model Predictive Control

OCP Optimal Control Problem

PLD Pegylated Liposomal Doxorubicin

RFPT Robust Fixed-Point Transformations

SAEM Stochastic Approximation of Expectation Maximization

List of symbols

$\alpha(\xi), \beta(\xi)$ nonlinear terms in the normal form of a continuous-time input affine nonlinear system

$\bar{t}_r, \sigma_r$ mean and standard deviation of a time duration corresponding to a relapse-remission cycle

$\bar{y}$ tumor volume normalization constant

$\bar{y}_p, \sigma_p$ mean and standard deviation of the difference between the peak tumor volume and the starting volume

$\bar{\boldsymbol{p}}$ normalization constants of the model parameter vector

$\boldsymbol{\Delta p}_k$ normalized error between the current, and previous model parameter estimations at time t_k

$\boldsymbol{\Delta y}_k$ normalized error between the measured and simulated tumor volumes up to the time instance t_k

$\boldsymbol{\delta}_s$ random effects of a mixed-effect model for time series s

$\boldsymbol{\theta}$ fixed-effects of a mixed-effect model

$\boldsymbol{\theta}_c$ vector containing the control parameters of the deform function

$\tilde{\boldsymbol{p}}_a^*$ lag one autocorrelation of each time-varying parameter

$\boldsymbol{\xi}$ transformed states of a continuous-time input-affine nonlinear dynamical system

$\boldsymbol{\xi}$ transformed states of a discrete-time nonlinear system

$\boldsymbol{d}^{\boldsymbol{c}}$ cumulative dosage associated with each administration

$\boldsymbol{d}$ sequence of administration doses

$\boldsymbol{g}$ path constraints of an OCP

$\boldsymbol{p_s}$ scaling constants for the parameter deviations

$\boldsymbol{p}_s^*$ estimated model parameters of subject s

$\boldsymbol{P}_{M_s}^*$ estimated time-varying parameter for each tumor volume measurement

$\boldsymbol{p}^{\mathrm{avg}}$ average model parameters used in the inverse model of the RFPT controller

$\boldsymbol{p}_c$ diagonal elements of R

$\boldsymbol{p}_k$ model parameters corresponding to the truncated time series $\boldsymbol{y}_k, \boldsymbol{u}_k$

$\boldsymbol{r}$ terminal constraints of an OCP

$\boldsymbol{s}_i$ artificial initial value used in the DMS

$\boldsymbol{w}$ optimization vector, containing states and inputs in the case of the DMS

$\boldsymbol{y}_k^W, \boldsymbol{u}_k^W$ previous W number of measurements at time t_k and all the inputs between the endpoints of the window

$\boldsymbol{y}_k, \boldsymbol{u}_k$ truncated versions of the vectors $\boldsymbol{y}_M, \boldsymbol{u}_N$ up to the time instance t_k

$\boldsymbol{y}_M, \boldsymbol{u}_N$ vectors, containing measurements and inputs at fixed times, respectively

Δt sampling time of discretization

δ convergence condition of the fixed-point iteration

$\ell_i(\boldsymbol{s}_i, d_i)$ integrated stage cost

κ_i shifting term of the ith administration in the bump function (2.5)

Λ control parameter of a linear controller defined in (3.39)

$\mathbb{X}, \mathbb{P}$ feasible sets of the states and parameters respectively

$\mathcal{E}(g)$ error term between the nonlinear term in the normal form of a system and the virtual input

$\mathcal{E}_d(g)$ error term between the nonlinear term in the discrete time normal form of a system and the virtual input

$\mathcal{G}(\mathcal{E}(g), g, \theta)$ deform function

$\mathcal{N}(0, \Sigma)$ normal distribution, centered at zero with covariance matrix Σ

$\mathcal{T}_c$ nonlinear coordinate transform of a nonlinear continuous-time input-affine dynamical system

$\mathcal{T}_d$ nonlinear coordinate transform of a discrete-time dynamical system

$\mathcal{U}(l_x, u_x)$ uniform distribution with bounds l_x and u_x

$\mathcal{V}_r, \mathcal{V}_o$ robust and optimal virtual populations

μ approximation term in the bump function (2.5)

$\mu_1, \mu_2, \mu_3, \mu_4, \mu_i^u$ scaled state and input variables in the ODE model

$\oslash$ element-wise division between two vectors

$\Phi(\boldsymbol{x}, u)$ state transition vector field of a nonlinear discrete time system

$\Psi(\boldsymbol{x}, y)$ inverse model of a discrete time nonlinear system

τ_i time of the ith drug administration

ε_{sk} noise on the k-th observed value of time series s

$a, b, n, w, ED_{50}, c, k_1, k_2, K_B$ model parameters, listed in Table 2.1

A_g, B_g, K_g Control parameters of the deform function (3.38)

d_i amount of the ith drug administration

d_{sum} total dose administered to a mouse

$E(\boldsymbol{x}(T))$ terminal cost of an OCP

f, g, h vector fields of a continuous-time input affine nonlinear system

f^*, g^*, h^* vector fields of a continuous-time input affine nonlinear system with perturbed parameters

g deformed variable

g^* fixed point for which $\mathcal{E}(g^*) = 0$

J_k kth optimization problem of the FIE

J_T cost function of an optimal control problem (OCP) on a fixed interval T

K gain matrix of a state-feedback controller

$L(\boldsymbol{x}(t), u(t))$ stage cost of an OCP

$L_f h$ Lie derivative of the vector field h along the vector field f

M number of measurements in a given time series

M_g number of fixed-point iterations

p p-value of the logrank test

p_{cl} closed-loop poles of the state-feedback controller

q relative degree of a dynamical system in Chapter 3

r control parameter in Section 4.2

R diagonal weighting matrix for the parameter deviation in (4.5)

$RMSE_s$ Root Mean Square Error of subject s

$S_{i,j}$ jth subject of the ith experiment

$S_{i,j}$ j-th subject of the i-th experiment

t_k time of the kth tumor volume measurement

$t_{survival}$ day on which mouse is terminated

u time function of the injection rate

$u_{\text{sum}}, e_{\text{sum}}$ total amount of drug administered and sum of the tracking error at the end of an experiment

v virtual input of an IO linearized continuous-time input affine nonlinear system

W number of measurements in the moving horizon of the MHE

X, U, Y state, input, and output spaces respectively

x_1 time function of the living tumor volume

x_2 time function of the dead tumor volume

x_3 time function of the drug level in the central compartment

x_4 time function of the drug level in the peripheral compartment

$x_{1c}, x_{2c}, x_{3c}, x_{4c}$ scaling constants of the states in the ODE model

y^{ref} tumor volume setpoint to be tracked

y_i value of the kth tumor volume measurement

y_{sk} kth observed value of time series s

Acknowledgements

The work conducted during my Ph.D. program would not have been possible without the extensive help, both professional and personal, that I have received throughout the past years of my life. First, I would like to express my sincerest gratitude to Dr. Levente Kovács, who introduced me to the research topic and provided a lot of major additional support throughout the years behind me. I would also like to thank the excellent guidance of Dr. Dániel András Drexler, who has provided me with a significant amount of professional advice during my research period and always pointed me toward the correct decisions. I believe that with their help, the final result of my work has a significantly higher value. I have also received great support from Dr. Gergely Szakács, Dr. András Füredi, and the whole staff at the Institute of Enzymology at the Eötvös Loránt Research Network, who provided me with the opportunity to in vivo validate the algorithms, designed in this work. In addition, I will also be grateful to Dr. Krisztián Kósi, whom I received a great deal of help throughout my undergraduate studies at the university. I am also thankful for the colleagues at the Physiological Controls Research Center of Óbuda University, who made the boring parts of the research colorful. On a personal side, I am obliged to my family and friends, who always considered the best for me and were always there. Their support was paramount during the past four years of my Ph.D. studies.

1

Introduction

The general treatment of cancerous diseases can be considered as one of the most challenging problems in modern medicine. In 2022, approximately 1.25 million people die from some variants of this illness in the European Union [R1]. While the numbers are decreasing due to better screening procedures, there has not been a single effective treatment developed that can stop the illness completely. The most commonly available treatment options are still surgery, radiotherapy, and chemotherapy. In order to improve the effectiveness of these therapies, individual treatment plans can be designed, which are tailored to the physiological and biological attributes of the patient. Since radiotherapy and chemotherapy also involve various side effects during the treatment period, individualization can also alleviate these negative consequences, thus improving the overall quality of life of the patient. In the case of chemotherapy, individualization means that one wishes to administer the least amount of drug during therapy that still ensures remission of the disease. Throughout my dissertation, I use methods from control theory to generate administration protocols for chemotherapy, using mathematical models that describe the treatment process. These models are able to predict the evolution of the tumor and can be used to calculate the optimal amount of drug that must be given to the patient. In addition, if one is able to regularly measure some aspects of the tumor that provide information on its evolution (e.g., its volume), the therapy can further be optimized, which leads to a closed-loop control scheme.

In the case of chemotherapy, another key issue is drug resistance, where the tumor adapts to the effect of a given drug and is no longer sensitive to each consecutive treatment. If the effect of drug resistance appears during treatment, the only viable option to continue the treatment is to change the applied chemotherapeutic agent, which can further put a strain on the already ill patient. Individualization of the treatment could also prolong the effectiveness of the given drug, which also justifies the aim of the current research [R2]. Metronomic therapy might also be a viable option to delay (or even avoid) drug resistance during the treatment, as reported in [R3]. In the case of metronomic chemotherapy, smaller doses are given to the patient more frequently, as opposed to conventional treatment schemes [R4]. Since the performance of a closed-loop control algorithm increases if the system can be controlled at any desired time instance, metronomic therapy can provide a better basis for the individualization.

Literature on chemotherapy optimization dates back to the beginning of the 1970s. Initial effort was put in the development of simulation models based on in vitro experiments and first principle assumptions [R5]. These models were then applied in the context of optimal control, for example as can be seen in [R6]. A survey on the results of the first ten years in the domain of optimal control of tumor growth models can be found in [R7]. These initial results heavily relied on explicit formulations of the optimal control problems, without the use of sophisticated numerical solvers. With the development of computational resources, researchers started to use numerical optimization algorithms to solve these optimal control problems. Moreover, soft computing techniques, and heuristics were also employed, which, together with the classical approaches, can be found in the survey by Shi et al. [R8] and Sbeity et al. [R9]. Some of the most recent results involve the use of Nonlinear Model Predictive Control (NMPC), Moving Horizon Estimation (MHE), and impulsive differential equations as can be seen in Chen et al. [R10] and Belfo et al. [R11].

In my work, I used two different strategies to control the tumor in a model-based manner. The first method was the Robust-Fixed-Point Transformations (RFPT) based nonlinear control algorithm, which is able to handle structural and parametric uncertainties of a nonlinear system. Since physiological control often involves both uncertainties, the RFPT was a promising candidate to solve the problem. While the method is less mature than other nonlinear robust techniques, including Sliding Mode Control, Adaptive Control, and Backstepping Control, it has several benefits that make it preferable. The first and foremost advantage lies in the simplicity of the controller design. The hardest part of the design is the computation of the inverse model, which can be automatically performed in Wolfram Mathematica. In the case of the other methods, a Lyapunov function is most often sought, which requires extensive trial and error computations. Additionally, the most robust method can either deal with structural or parametric uncertainties but not both, which makes the current choice even more attractive. Aside from using the RFPT method for control design in this dissertation, I also wanted to contribute to its theoretical development since I believe the method can be a viable alternative to other existing robust techniques based on prior experience.

The second approach was the use of Nonlinear Model Predictive Control (NMPC) and Moving Horizon Estimator(MHE), which are standard, optimization based control and estimation algorithms in the field of nonlinear control. While robustness is, in general, harder to ensure in the case of NMPC, the method permits the incorporation of various constraints during the optimization in a simple way. Therefore, accounting for the benefits and drawbacks of both approaches, I have decided to test both in my dissertation.

1.1 Robust Fixed-Point Transformation

The idea of using fixed-point iteration in nonlinear control was introduced by Tar and Rudas [R12]. A key motivation behind the development of the control method was to provide an alternative to the use of Lyapunov functions in adaptive control. The RFPT uses a linear controller to produce a reference trajectory, which is then further manipulated by a deform function. The deform function induces a fixed-point iteration, which can mitigate the effect of structural and parametric uncertainties between the plant and its inverse model. The role of the inverse model is to compute the input signal of the system corresponding to a certain desired reference trajectory, which is produced by the linear controller. The closed-loop block diagram is illustrated in Figure 1.1. Initial directions in the research focused on the development of general deform functions that can be applied to a wide variety of nonlinear systems as can be seen in [R14, R13] and [R12]. Multiple-input multiple-output extensions of the algorithm have been also considered in [R15] and [R16]. Preliminary results on the stability of the controller were also discussed in [R17] and [R18].

There is a wide variety of application examples ranging from mechanical systems ([R20, R19]) to problems in physiological control as well ([R21]). A study on the implementation of the algorithm on a double-rotor test bench can be found in [R22].

1.2 Nonlinear Model Predictive Control

Nonlinear Model Predictive Control (NMPC) is an optimization-based method in the context of nonlinear control [R23, R24]. At each control time instance, the controller optimizes a cost function through a sequence of control actions that alter the behavior of the simulated system. The optimal solution then minimizes some quantifiable goals incorporated in the cost function and also complies with constraints that are imposed on the optimization problem. The control action at the current time is then the first element of the vector of control variables, and the process is repeated at the next control time instance. The cost function often contains terms that penalize the control goal and the control effort in some sense. In the case of therapy optimization, the control goal is to force the tumor volume to be as small as possible, while the control effort is characterized by the amount of drug given during the prediction, which should also be minimal.

1.3 Moving Horizon Estimation

NMPC requires state estimates of the underlying process to be able to predict the evolution of the system during the optimization of the doses. In the case of chemotherapy optimization, there is a vastly limited number of measurement channels that are available, thus state estimation is a necessary part of the algorithm.

Moving Horizon Estimation is a state estimation technique for nonlinear systems, which can be considered to be a dual problem of the NMPC. It is both capable of the estimation of time-varying parameters and the corresponding state estimates [R23]. The goal in the case of MHE is to find an optimal estimate of the current state of the process using a fixed number of previous measurements with respect to a cost function. Akin to its counterpart, MHE is also an optimization-based technique that uses the dynamical model of the system to provide additional insight into the observed process. The strategy is also capable of handling time-varying parameters of a given model, thus proving to be a reasonable candidate for capturing the adaptation strategy of the disease to the drug. Furthermore, MHE has a special case called the Full Information Estimator, where the window size of the estimator is equal to the number of every available measurement. This can be useful for estimating model parameters and tuning the MHE which will be demonstrated later in the work.

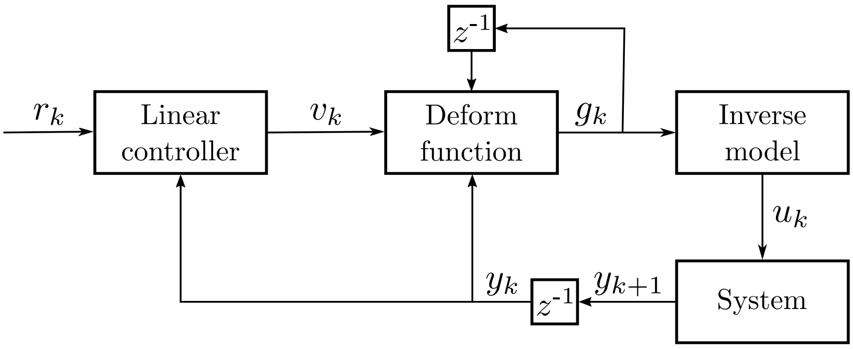


Figure 1.1: Block diagram of the RFPT method in discrete time. The variable r_k denotes the reference trajectory, v_k is the trajectory corrected by a linear control law, g_k is the output of the deform function, u_k is the input, and y_{k+1} is the response of the nonlinear system.

1.4 Experimental validation and iterative design

The optimal control algorithms designed in my dissertation were tested in vivo through mice experiments. Each experiment provided additional insight that could be used to improve the algorithms. As a consequence, multiple different implementations of the MHE-NMPC scheme were carried out, and these were further tested with new experiments. In the case of the NMPC, the central issue that had to be solved was that the algorithm often encountered convergence issues at different optimization time instances. It became clear during my research that this issue primarily originated from both the scaling of the state variables and the large variation between the model parameters. In the last iteration of the controller design, the convergence issues were completely eliminated by modifying the cost function of the optimization problem and obtaining additional experimental data, which improved the accuracy of the model parameters.

In Figure 1.2. a flowchart is presented, which shows the components of the algorithm in its current state. Moreover, the figure serves as a blueprint on how the algorithm can be fine tuned with additional experimental data in the future. After a finished experiment $\boldsymbol{S}_N$, all the experimental data are aggregated $\boldsymbol{S} = \{\boldsymbol{S}_0,\ \ldots,\ \boldsymbol{S}_N\}$, which is then used in the FIE to compute initial model parameters $\boldsymbol{P} = \{\boldsymbol{p}_0^*,\ \ldots,\ \boldsymbol{p}_M^*\}$, where M is the number of subjects from all the experiments. These initial model parameters are then used in the SAEM algorithm to create mixed-effect parameters with a fixed-effect vector $\boldsymbol{\theta}$ and covariance matrix Σ. Using the updated $\boldsymbol{\theta}$, the MHE is tested on all the experimental data. Additionally, using $\boldsymbol{\theta}$, and Σ two virtual populations are generated to tune the gain of the NMPC controller, and to validate the combined MHE-NMPC algorithm in silico. If the in silico validation is successful, the algorithm can be tested in vivo.

1.5 Chronological order of the research

The contents of the thesis were organized such that parts of the control design were logically separated, instead of following a chronological order of the research. Nevertheless, in order to clarify certain design decisions, the timeline of the research should also be presented. The following list describes certain phases of the work:

1. Controller design and methodological research on the RFPT controller, using data from [R25]. In silico validation of the method using the model with three state variables [C1, C2, C3, C4].

2. Implementation of the first iteration of the NMPC controller to the model with three state variables, in silico and in vivo validation of the design using the same parameter set as in the case of the RFPT [C5, C6].

3. Implementation of the second iteration of the NMPC controller with the virtual population-based tuning rule. Utilization of derivative-free optimizers and constraint on the cumulative toxicity. Experimental validation of the design, using the previous MHE implementation with minor changes [C7].

4. Complete redesign of the MHE, and the implementation of the FIE. Introduction of the scaling terms in the NMPC and the complete elimination of all numerical issues regarding its convergence. Experimental validation of the new controller without initial identification dose [C8].

1.6 Outline of the thesis

In Chapter 4.1.3., the tumor growth models are presented, with different possible dosing schemes. Chapter 3. describes the mathematical formulation of the RFPT approach. In particular, the basics of the Input-Output linearization are presented, from which different variants of the RFPT approach were derived and tested in silico. In Chapter 4., I present the parameter estimation problem, and introduce the FIE and the MHE respectively, with the obtained parameters calculated from the experiments. The theory of the NMPC is presented along with the different iterations of the design with their in silico validation. In Chapter 5., the in vivo experiments are presented. The conclusions are drawn in Chapter 6.

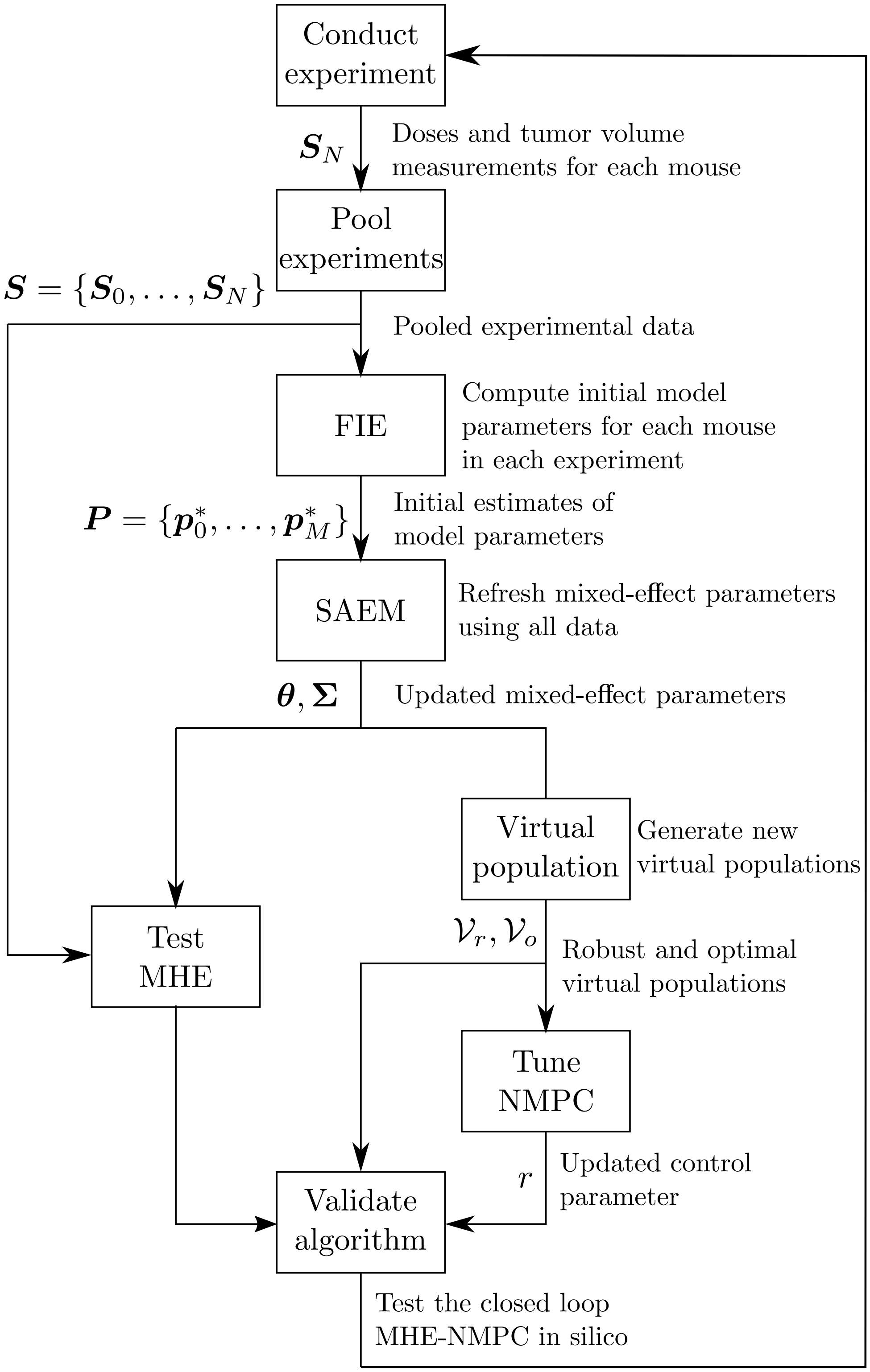

Conduct experiment
S_N
Doses and tumor volume measurements for each mouse
Pool experiments
$S = \{S_0, \dots, S_N\}$
Pooled experimental data
FIE
Compute initial model parameters for each mouse in each experiment
$P = \{p_0^*, \dots, p_M^*\}$
Initial estimates of model parameters
SAEM
Refresh mixed-effect parameters using all data
θ, Σ
Updated mixed-effect parameters
Virtual population
Generate new virtual populations
Test MHE
$\mathcal{V}_r, \mathcal{V}_o$
Robust and optimal virtual populations
Tune NMPC
r
Updated control parameter
Validate algorithm
Test the closed loop MHE-NMPC in silico

Figure 1.2: Flowchart of the iterative design for the NMPC-MHE approach.

2

Tumor growth models

The control algorithms presented in the dissertation require a model of the controlled system. Such a model should be able to describe the dynamics of the tumor in conjunction with the administration and elimination of the drug. Hence, a tumor model is discussed in this chapter, which is able to capture these aspects of the biological process.

There is a large variety of models which captures key mechanisms of tumor growth, demonstrated in the survey of [R26]. In my work, I was primarily focusing on the use of ordinary differential equation-based models, which are useful in the description of time series that are dependent on a single independent variable. Moreover, one can make a distinction between macroscopic and microscopic models that describe the evolution of the tumor [R27]. In the case of microscopic models, interactions are modeled at the cell level, which often entails a complex model with a significant number of state variables and model parameters. Since the current technology only permits the measurement of a limited number of properties of the tumor, identification of the parameters of these models is difficult or even intractable in many cases [R28]. By contrast, macroscopic models describe the overall behavior of the system, only using a few state variables, which is suitable for estimation and control design. This loss of complexity, however, limits the predictive power of the model and thus the use of time-varying model parameters has to be considered.

In my dissertation, the tumor growth model describes the evolution of breast cancer in a clinically relevant, genetically engineered mouse model. In this mouse model Brca1, a DNA repair gene, and p53 a regulator of cell cycle and genome stability, were knocked out in breast epithelial cells. Mammary tumors obtained in this way highly resemble the Brca1-linked, triple-negative, hereditary breast cancer in humans, since their molecular, immunohistochemical, morphological, and genetic characteristics are almost indistinguishable from their human counterparts [R29]. These tumors also respond to chemotherapy very similarly, as an initial treatment with doxorubicin, docetaxel or cisplatin significantly reduce tumor size and induce remission. As a consequence, findings obtained by using this mouse model are frequently translated to human cancer clinics due to their similarity to human breast cancer.

Unfortunately, long-term therapy often fails due to the emergence of drug resistance [R31, R30], and most of the novel therapeutic approaches to tackle it are in the early developmental phase [R34, R33, R32]. The chemotherapeutic drug used in the experiments was pegylated liposomal doxorubicin (PLD). While it was demonstrated in previous works that the drug increases relapse-free and overall survival by 6- and 3-fold respectively, these tumors cannot be cured using conventional chemotherapy regimens [R25].

Based on the previous biological considerations, the input of the system is the amount of PLD that is administered to the subjects via injections at a given day τ_k [day]. The time of administrations are denoted with $\tau_i \in \mathbb{R}^+$, $i \in \mathbb{N}_0$ for which the ordering $\tau_0 < \tau_1 < \cdots <$

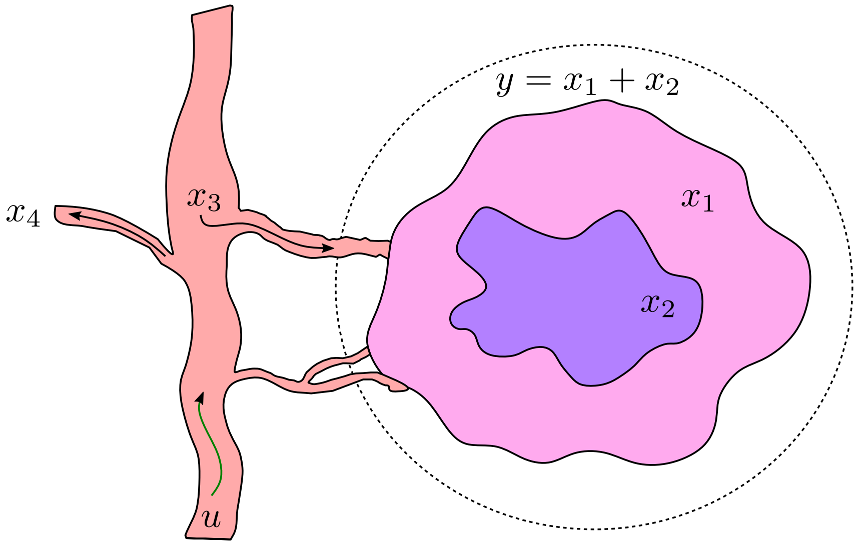


Figure 2.1: Visualization of the model. Here, x_1 is the living tumor volume (pink region),x_2 is the dead tumor volume (purple region), x_3 is the concentration of drug in the blood (red region), x_4 is the concentration of the drug in the peripheral compartment, and u denotes the rate of injection.

$\tau_i < \cdots < \tau_N$ holds, and the amount of the drug is denoted by $d_i = d(\tau_i) \in \mathbb{R}^+$ [mg/kg]. In practice, the size of the tumor implanted under the skin of the mice can be measured using calipers during the experiment. By measuring the width and length of the tumor, its volume $y \in \mathbb{R}^+$ can be approximated as

$$y_k = y(t_k) = \frac{\pi}{3}(\text{length}(t_k) \cdot \text{width}(t_k))^{\frac{3}{2}}, \tag{2.1}$$

according to the formula derived in [R35], where $t_k \in \mathbb{R}^+$, $k \in \mathbb{N}_0$ [day] is the measurement time instance, and $t_0 < t_1 < \cdots < t_k < \cdots < t_M$. This quantity will be the only output of the system, since taking measurements from other channels is technologically or economically infeasible currently.

Throughout my dissertation, I used two macroscopic tumor growth models, adapted from [R37, R36], which describe the tumor dynamics, pharmacokinetics, and pharmacodynamics of the drug. The first model has the form

$$\begin{aligned} \dot{x}_1 &= (a-n)x_1 - b\frac{x_1 x_3}{ED_{50} + x_3} \\ \dot{x}_2 &= nx_1 + b\frac{x_1 x_3}{ED_{50} + x_3} - wx_2 \\ \dot{x}_3 &= -c\frac{x_3}{K_B + x_3} + u \\ y &= x_1 + x_2 \end{aligned} \tag{2.2}$$

where $x_1 \in \mathbb{R}^+$ is the living tumor volume in [mm^3], $x_2 \in \mathbb{R}^+$ is the dead volume in [mm^3], $x_3 \in \mathbb{R}^+$ is the level of drug in the host measured in [mg/kg] and $u \in \mathbb{R}^+$ is the injection rate in [mg/kg/day].

Further research showed that by augmenting the model with a fourth state, that relates to the drug concentration in a peripheral compartment, the model can describe the pharmacokinetics of the drug more accurately [R36]. The augmented model is

$$
\begin{aligned}
\dot{x}_1 &= (a-n)x_1 - b\frac{x_1 x_3}{ED_{50} + x_3} \\
\dot{x}_2 &= nx_1 + b\frac{x_1 x_3}{ED_{50} + x_3} - wx_2 \\
\dot{x}_3 &= -(c+k_1)x_3 + k_2 x_4 + u \\
\dot{x}_4 &= k_1 x_3 - k_2 x_4 \\
y &= x_1 + x_2,
\end{aligned}
\tag{2.3}
$$

where $x_3 \in \mathbb{R}^+$ now is the time function of the drug level in the central compartment [mg/kg], and $x_4 \in \mathbb{R}^+$ is the time function of the drug level in the peripheral compartment [mg/kg]. The input $u \in \mathbb{R}^+$ is the time function of the injection rate [mg/kg/day], while the output $y \in \mathbb{R}^+$ [mm^3] is given as the total tumor volume. A description of the model parameters with their corresponding dimensions can be seen in Table 2.1. and simple visualization of the system is depicted in Figure 2.1.

An important property of both systems is positivity, which means that all the states, the input, the output, and the model parameters have to be strictly positive for the model. This has two immediate consequences in the control design case: the input signal must be constrained to be positive (i.e., no drug can be extracted from the body), and the corresponding state trajectory must be positive for all times (resulting in positive tumor volumes).

Both models can be continuously controlled through the input variable u, which must be transformed into a sequence of impulses to mimic realistic administration schemes via a series of injections. One possibility is to omit the term u from the differential equations and apply the dose directly to the corresponding state variable x_3, which leads to an impulsive differential equation [R38]. Each τ_i introduces an interval $(\tau_0, \tau_1], (\tau_1, \tau_2], \dots, (\tau_i, \tau_{i+1}], \dots$, on which the model is piece-wise defined with the rule

$$
\boldsymbol{x}(\tau_i^+) = \boldsymbol{x}(\tau_i^-) + [0\ 0\ 1\ 0]^\mathsf{T}\, d_i. \tag{2.4}
$$

The impulsive effects are characterized by $\boldsymbol{x}(\tau_i^+) = \lim_{h\to 0^+} \boldsymbol{x}(\tau_i + h)$ in conjunction with $\boldsymbol{x}(\tau_i^-) = \lim_{h\to 0^+} \boldsymbol{x}(\tau_i - h)$. Equation (2.4) means that the system is simulated from τ_i to τ_{i+1} with the initial condition that is the endpoint of the previous simulation, modified with the input value d_i.

My initial approach was to represent the impulses using Dirac delta approximations by redefining the input in (2.2) such that

$$
u = u(t, \boldsymbol{\tau}, \boldsymbol{d}) = \begin{cases} \dfrac{d_i}{2\mu}\left(1 + \cos\left(\dfrac{\pi(t-\kappa_i)}{\mu}\right)\right), & \tau_i \le t \le \tau_i + 2\mu \\ 0, & \tau_i + 2\mu < t < \tau_{i+1} \end{cases} \tag{2.5}
$$

where $\boldsymbol{\tau} \in \mathbb{R}^N$ contains the input time instances, $\boldsymbol{d} \in \mathbb{R}^N$ contains the corresponding doses, μ controls the approximation, $\kappa_i = \tau_i + \mu$ is the shifting term and $t \in [\tau_0, \tau_N]$ [C5]. This is a compactly supported smooth approximation of the Dirac delta distribution where the smoothness property could be favorable when using gradient-based optimizers. In simple terms, this formalism essentially places a Dirac impulse in the beginning of each interval $(t_i, t_{i+1}]$. Since the integral of (2.5) with $d_i = 1$ is one, d_i can be interpreted as the amount of drug that is injected into the patient directly. In order to determine the approximation parameter, it can be taken into account that one administration takes approximately 15 seconds in mice, which yields the choice $\mu = (15/86400)/2$ [day].

In order to use the model for predictive purposes, its parameters have to be estimated from time series data. During the estimation, a number of issues must be addressed which

Table 2.1: Model parameters and their dimensions. Each model parameter is a positive real number.

Parameter	Unit	Description
a	1/day	Tumor growth rate
n	1/day	Tumor necrosis rate
b	1/day	Drug efficacy rate
ED_{50}	mg/kg	Median effective dose
w	1/day	Dead tumor cell washout
c	1/day	Clearance rate of the drug
k_1	1/day	Pharmacokinetic parameter
k_2	1/day	Pharmacokinetic parameter
K_B	mg/kg	Michaelis-Menten constant

stem from the lack of data, inter-patient variation in the parameters, and technical difficulties in the estimation algorithm (e.g., proper choice of the initial parameter value of the estimation).

If repeated experimental data is available on the same process, a standard estimation technique is to consider the parameters to have mixed effects, which consist of fixed effects that describe the population in an average manner, and random effects that pertain to each individual sample. By using mixed effects, the inter-patient correlations can be meaningfully represented by an associated probability distribution. Moreover, the increased number of data that is taken into account during the estimation entails more precise parameter values.

In previous works, the Stochastic Approximation of Expectation Maximization (SAEM) was used to obtain mixed-effect parameters for the model [R36]. While it is possible to fit the model to the data using the method, the quality of the estimation often remains unsatisfactory. Since SAEM is an iterative algorithm, this problem can be largely attributed to the difficulty of finding a proper initial parameter vector for the estimation. Furthermore, the estimation algorithm does not allow the use of parameter constraints, which may result in unexpected behavior in the model. For example, one might obtain a parameter set that ensures a good fit on the measurement data, but the parameter c is so small that there is essentially no depletion of the drug, which is physiologically unreasonable. In Chapter 4, a solution to these problems is presented, and time-varying parameters will also be considered.

3

Robust Fixed-Point Transformations

Thesis 1: Model-based control using the RFPT approach

Thesis 1
I contributed to the theoretical description of the RFPT method by connecting it with the Input-Output (IO) linearization principle. Moreover, I developed a functional fixed-point iteration based variant of the algorithm in conjunction with a purely discrete-time version. I also tested the viability of each different strategy in silico on a tumor growth model.

Publications relevant to the theses: [C3, C2, C1].

My initial research objective was to prove that the system can be controlled continuously and to investigate the robustness of the given algorithm. For this reason, I designed various RFPT-based controllers, which I tested on the previously introduced models. Moreover, I also connected the methodology to the IO linearization principle in a rigorous manner, from which a number of possible implementations of the base idea emerged.

3.1 Continuous time Input-Output linearization

Consider the following nonlinear input-affine dynamical system

$$\begin{aligned} \dot{\boldsymbol{x}} &= f(\boldsymbol{x}) + g(\boldsymbol{x})u \\ y &= h(\boldsymbol{x}) \end{aligned} \tag{3.1}$$

where $\boldsymbol{x}(t) \in \mathbb{R}^n$ is the state vector at time t, $u(t) \in \mathbb{R}$ is the system input at time t, while $y(t) \in \mathbb{R}$ is the output of the system at time t, where $t \in \mathbb{R}^+$, and f, g, h are smooth vector fields. Differentiating the output leads to

$$\dot{y} = \frac{\partial h}{\partial \boldsymbol{x}}\dot{\boldsymbol{x}} = \frac{\partial h}{\partial \boldsymbol{x}}[f(\boldsymbol{x}) + g(\boldsymbol{x})u] \triangleq L_f h + L_g h\ u \tag{3.2}$$

where $L_f h$ and $L_g h$ are the Lie derivatives of h along the vector fields f and g, respectively. The notation admits the property [R39]

$$\begin{aligned} L_f^0 h &= h \\ L_f^k h &= L_f L_f^{k-1} h = \frac{\partial (L_f^{k-1} h)}{\partial \boldsymbol{x}} f, \quad k \geq 1. \end{aligned} \tag{3.3}$$

One can assume that the control input u appears first in the qth derivative of the output which poses the condition

$$L_g L_f^{q-1} h = \frac{\partial (L_f^{q-1} h)}{\partial \boldsymbol{x}} g \neq 0. \tag{3.4}$$

where q is called the relative degree of the system.

Definition 1 (Relative degree)**.** *The nonlinear system* (3.1) *has relative degree* q $(1 \leq q \leq n)$ *in a region* $D_0 \subset D$*, if*

$$\begin{aligned} &L_g L_f^{i-1} h(\boldsymbol{x}) = 0, \ i = 1, .., q-1 \\ &L_g L_f^{q-1} h(\boldsymbol{x}) \neq 0 \end{aligned} \tag{3.5}$$

for all $\boldsymbol{x} \in D_0$.

I also assumed that the relative degree equals the state dimension, i.e., $q = n$, which involves no zero dynamics in the normal form of the system. While this choice limits the generality of the presented method, the exposition becomes simpler with fewer technical details. A comprehensive work on the treatment of zero dynamics can be found in [R40]. based on this assumption, one can define the following coordinate transformation

$$\begin{pmatrix} \xi_1 \\ \xi_2 \\ \vdots \\ \xi_q \end{pmatrix} = \mathcal{T}_c(\boldsymbol{x}) = \begin{pmatrix} h(\boldsymbol{x}) \\ L_f h(\boldsymbol{x}) \\ \vdots \\ L_f^{q-1} h(\boldsymbol{x}) \end{pmatrix} \tag{3.6}$$

where $\mathcal{T}_c : D_{\boldsymbol{x}} \subset \mathbb{R}^n \to D_{\boldsymbol{\xi}} \subset \mathbb{R}^n$ must be a diffeomorphism. This transformation maps the system to the normal form

$$\begin{aligned} \dot{\xi}_1 &= \xi_2 \\ \dot{\xi}_2 &= \xi_3 \\ &\vdots \\ \dot{\xi}_{q-1} &= \xi_q \\ \dot{\xi}_q &= \alpha(\boldsymbol{\xi}) + \beta(\boldsymbol{\xi}) u \\ y &= \xi_1. \end{aligned} \tag{3.7}$$

In its normal form, the system can be divided into two distinct parts. There is a linear part that contains a chain of integrators, and a single nonlinear equation with the terms

$$\begin{aligned} \alpha(\boldsymbol{\xi}) &= [L_f^q h(\boldsymbol{x})]_{\boldsymbol{x} = \mathcal{T}_c^{-1}(\boldsymbol{\xi})} \\ \beta(\boldsymbol{\xi}) &= [L_g L_f^{q-1} h(\boldsymbol{x})]_{\boldsymbol{x} = \mathcal{T}_c^{-1}(\boldsymbol{\xi})}. \end{aligned} \tag{3.8}$$

Using the nonlinear equation in (3.7) the input is defined as

$$u = \frac{v - \alpha(\boldsymbol{\xi})}{\beta(\boldsymbol{\xi})} \tag{3.9}$$

which cancels out the nonlinearities, hence $\dot{\xi}_q = v$, where v is called the virtual input. Combining equations (3.7) and (3.9) one obtains a linear system

$$\begin{aligned} \dot{\boldsymbol{\xi}} &= A\boldsymbol{\xi} + Bv \\ y &= C\boldsymbol{\xi}, \end{aligned} \tag{3.10}$$

which can be stabilized in the origin by the static state feedback $v = -K\boldsymbol{\xi}$, using pole placement to determine the gain matrix K.

3.2 Continuous RFPT

Suppose that equation (3.9) is derived from an imprecise model with vector fields f^*, g^*, h^*, which are structurally identical to f, g, h in (3.1). It is further assumed that no bifurcation is caused by parametric uncertainties so that the equilibrium properties of the system remain the same. Using (3.7) and (3.9) one obtains

$$\dot{\boldsymbol{\xi}}_q = \alpha(\boldsymbol{\xi}) + \beta(\boldsymbol{\xi})\left(\frac{v - \alpha^*(\boldsymbol{\xi})}{\beta^*(\boldsymbol{\xi})}\right) \tag{3.11}$$

with imprecise terms in the inverse

$$\begin{aligned} \alpha^*(\boldsymbol{\xi}) &= [L^q_{f^*} h^*(\boldsymbol{x})]_{\boldsymbol{x}=\mathcal{T}_c^{-1}(\boldsymbol{\xi})} \\ \beta^*(\boldsymbol{\xi}) &= [L_{g^*} L^{q-1}_{f^*} h^*(\boldsymbol{x})]_{\boldsymbol{x}=\mathcal{T}_c^{-1}(\boldsymbol{\xi})}. \end{aligned} \tag{3.12}$$

In (3.11), one can see that the nonlinearity of the system becomes more complex since the inverse model does not cancel out the relevant terms. This means, that the control law produced by the linear controller not necessarily stabilizes the nonlinear system. In order to circumvent this issue, the following error term is defined [C2]

$$\mathcal{E}(g) = \alpha(\boldsymbol{\xi}) + \beta(\boldsymbol{\xi})\left(\frac{g - \alpha^*(\boldsymbol{\xi})}{\beta^*(\boldsymbol{\xi})}\right) - v \tag{3.13}$$

where v is given by the state feedback controller, g is a new input that will be manipulated so that (3.13) is satisfied and $\mathcal{E}(\boldsymbol{\xi}, g, v) = \mathcal{E}(g) : \mathbb{R}^q \times \mathbb{R} \times \mathbb{R} \to \mathbb{R}$. The task is to find a solution g^*, which forces $\mathcal{E}(g)$ to be zero, thus ensuring that $\dot{\boldsymbol{\xi}}_q = v$. The solution can be found using fixed-point iterations, which is a standard method utilized in root-finding problems. First, the equation $\mathcal{E}(g) = 0$ is converted into a fixed-point form, from which the fixed-point iteration is given by

$$\mathcal{E}(g) + g = g \to \mathcal{E}(g_i) + g_i = g_{i+1}, \tag{3.14}$$

where $i \in \mathbb{N}^+$ is an iteration index, independent of the sampling time. This is a functional equation in $g(t)$, where the solution satisfies $\mathcal{E}(g_*) = g_* = 0$. In order to assess the convergence of the iteration, using the Banach Fixed-Point Theorem, one has to define k-contractive operators [R41].

Definition 2 (k-contractivity). *An operator $F : M \subseteq X \to X$ on a metric space (X, d) is called k-contractive on M if*

$$d(F(x), F(y)) \leq kd(x, y) \tag{3.15}$$

with fixed $0 \leq k < 1$ and for all $x, y \in M$.

Contractivity of continuous operators can be verified by the following condition [R41]

$$||F'(g)|| \leq k < 1. \tag{3.16}$$

If the operator F is k-contractive, the Banach Fixed-Point Theorem guarantees the convergence of the associated iteration.

Theorem 1 (Banach Fixed-Point Theorem). *If a given operator $F : M \subseteq X \to X$ is k-contractive on M then F admits a unique fixed point on M, i.e., $F(r_*) = r_*$, and r_* can be obtained by successive iterations in the form of $F(r_n) = r_{n+1}$ for an arbitrary $r_0 \in M$.*

In the current scenario, $g \in M \subseteq X$ lies in the Banach space of Lebesgue measurable functions on a finite interval $[0, t]$ with finite essential supremum $X = \mathcal{L}^{\infty}([0, t], \mathbb{R})$, and the operator is $F : \mathcal{E}(g) + g$. Consequently, if

$$\left|\left|\frac{\mathrm{d}\mathcal{E}(g)}{\mathrm{d}g} + 1\right|\right| < 1 \tag{3.17}$$

holds, then the parametric errors can be eliminated between the plant and the nominal model. However, in the vast majority of cases, the condition does not hold due to the nature of the error term $\mathcal{E}$. In order to modify the inherent behavior of $\mathcal{E}$, a deform function can be introduced as

$$\mathcal{G}(\mathcal{E}(g_i), g_i, \boldsymbol{\theta}_g) = g_{i+1} \tag{3.18}$$

with the property

$$\mathcal{G}(\mathcal{E}(g_*), g_*, \boldsymbol{\theta}_g) = g_* \to \mathcal{E}(g_*) = 0 \tag{3.19}$$

whence $\mathcal{G}(\mathcal{E}(g), g, v, \boldsymbol{\theta}) = \mathcal{G}(g) : \mathbb{R} \times \mathbb{R} \times \mathbb{R} \times \mathbb{R}^m \to \mathbb{R}$ is the deform function, with initial condition g_0, parametrized by $\boldsymbol{\theta}_g \in \mathbb{R}^m$, where m is the number of control parameters. Once the structure of the deform function can handle the nonlinearities, proper choice of $\boldsymbol{\theta}_g$ will render the iteration (3.18) convergent due to the Banach fixed point theorem. In practice, only a fixed number of iterations can be performed, such that $i \in \{0, \ldots, M_g\}$. The closed loop system is then defined by the following set of differential equations [C2]

$$\begin{aligned}
\dot{\xi}_1 &= \xi_2 \\
\dot{\xi}_2 &= \xi_3 \\
&\vdots \\
\dot{\xi}_{q-1} &= \xi_r \\
\dot{\xi}_q &= \mathcal{E}(g_{M_g}) + v \\
g_{i+1} &= \mathcal{G}(\mathcal{E}(g_i), g_i, \boldsymbol{\theta}_g) \\
v &= -K\boldsymbol{\xi} \\
y &= \xi_1.
\end{aligned} \tag{3.20}$$

The initial condition of $\boldsymbol{\xi}(0)$ is problem dependent, while g_0 can be determined by supposing that no parametric uncertainty is present, from which $g_0 = v$.

3.3 Discretization and real-time iteration

Eliminating the parametric errors in $\mathcal{E}(g)$ can also be performed online by discretizing equation (3.11) [C1]. Using Euler discretization (or any other numerical method without loss of generality) with sampling time $\Delta t > 0$ leads to

$$\xi_q^+ = \xi_q + \Delta t \left[\alpha(\boldsymbol{\xi}) + \beta(\boldsymbol{\xi}) \left(\frac{v - \alpha^*(\boldsymbol{\xi})}{\beta^*(\boldsymbol{\xi})}\right)\right] \tag{3.21}$$

where $\boldsymbol{\xi} := \boldsymbol{\xi}(k\Delta t)$, $v := v(k\Delta t)$ and $\boldsymbol{\xi}^+ := \boldsymbol{\xi}((k+1)\Delta t), k \in \mathbb{N}_0$. To render the system linear, one can postulate a similar requirement to (3.13), namely

$$\mathcal{E}(g) = \alpha(\boldsymbol{\xi}) + \beta(\boldsymbol{\xi}) \left(\frac{g - \alpha^*(\boldsymbol{\xi})}{\beta^*(\boldsymbol{\xi})}\right) - v \tag{3.22}$$

where $g := g(k\Delta t)$. Because of the discretization, the associated fixed-point iteration is no longer function, but scalar-valued, hence $g \in \mathbb{R}$. Since the previous observations on fixed-point convergence of $\mathcal{E}(g)$ holds here as well, the deform function (3.18) must be included in the closed-loop to ensure proper convergence:

$$\begin{aligned}\xi_1^+ &= \xi_1 + \Delta t \xi_2 \\ &\vdots \\ \xi_q^+ &= \xi_q + \Delta t \mathcal{E}(g) \\ g^+ &= \mathcal{G}(\mathcal{E}(g) - v, g, \boldsymbol{\theta}_g) \\ v &= -K\xi \\ y &= \xi_1.\end{aligned} \tag{3.23}$$

If the requirement

$$\left|\left|\frac{\mathrm{d}\mathcal{G}(\mathcal{E}(g), g, \boldsymbol{\theta}_g)}{\mathrm{d}g}\right|\right| < 1 \tag{3.24}$$

can be ensured in each k step, the sequence $g \to g^*$ could render the system to be linear in a fixed number of steps, whence the state feedback can stabilize the system in the origin. The initial condition of g_0 can be determined by supposing that no parametric uncertainty is present, from which $g(0) = v(0)$.

3.4 Discrete-time input-output linearization

I have also carried out a purely discrete-time variant of the algorithm. In the discrete-time scenario, a more general approach must be considered, since the result of the discretization of a nonlinear differential equation is seldom input-affine. A compact, but detailed description of the discrete linearization process can be found in [R42]. Consider the following nonlinear discrete-time dynamical system:

$$\begin{aligned}\boldsymbol{x}^+ &= \Phi(\boldsymbol{x}, u) \\ y &= h(\boldsymbol{x}),\end{aligned} \tag{3.25}$$

where $\boldsymbol{x} := \boldsymbol{x}(k) \in X \subset \mathbb{R}^n$ is the state vector at time step k with forward shift operator $\boldsymbol{x}^+ = \boldsymbol{x}(k{+}1) \in X \subset \mathbb{R}^n$, $u(k) \in U \subset \mathbb{R}$ is the input at time step k, $y(k) \in Y \subset \mathbb{R}$ is the output at time step k, where $k \in \mathbb{N}_0$, and both X and U are open and connected, containing the origin. Moreover, $\Phi(\boldsymbol{x}, u)$ is a smooth vector field on $X \times U$ and $h(\boldsymbol{x})$ is a smooth scalar field on X. In the continuous case, the underlying idea of the IO linearization principle was to differentiate the output of the system until the input signal explicitly appears in the rth derivative of the output. In the discrete-time case, rather than using the differential operator, the composition operator must be employed. Thus, the vector field $\Phi(\boldsymbol{x}, u)$ is iterated in its first argument r times, then composed with the output function $h(\boldsymbol{x})$. The iteration of Φ is then defined as

$$\begin{aligned}h^0(\boldsymbol{x}, u) &= h(\boldsymbol{x}) \\ h^1(\boldsymbol{x}, u) &= h \circ \Phi(\boldsymbol{x}, u) \\ &\vdots \\ h^j(\boldsymbol{x}, u) &= h \circ \Phi(\Phi^{j-1}(\boldsymbol{x}, u), u)\end{aligned} \tag{3.26}$$

which is precisely equal to the output delayed by j steps, i.e.,

$$y(k+j) = h^j(\boldsymbol{x}, u). \tag{3.27}$$

In the discrete case, the relative degree q means that at the kth time step, the control signal u will explicitly alter the system after q consecutive steps. Based on the previous considerations, one can obtain the following relation:

$$\frac{\partial}{\partial u} h^j(\boldsymbol{x}, u) = \frac{\partial h(\boldsymbol{x})}{\partial \boldsymbol{x}} \left(\frac{\partial \Phi(\boldsymbol{x}, u)}{\partial \boldsymbol{x}} \right)^{j-1} \frac{\partial \Phi(\boldsymbol{x}, u)}{\partial u} \tag{3.28}$$

from which the definition of the discrete-time relative degree arises.

Definition 3 (discrete-time relative degree)**.** *The discrete-time nonlinear system (3.25) has a relative degree* $1 \leq q \leq n$ *in a region* $D_0 \subseteq D$ *if*

$$\min_{q \in \mathbb{N}} \frac{\partial h(\boldsymbol{x})}{\partial \boldsymbol{x}} \left(\frac{\partial \Phi(\boldsymbol{x}, u)}{\partial \boldsymbol{x}} \right)^{q-1} \frac{\partial \Phi(\boldsymbol{x}, u)}{\partial u} \neq 0 \tag{3.29}$$

for all $\boldsymbol{x} \in D_0$.

If the relative degree of the system is finite, it implies that the relation

$$y(k+q) = h^q(\boldsymbol{x}, u) \tag{3.30}$$

is locally solvable in u by the inverse function theorem, thus

$$u = \Psi(\boldsymbol{x}, y(k+q)) \tag{3.31}$$

where $\Psi(\boldsymbol{x}, y(k+q))$ is the inverse model and assumed to be well-defined and unique on $X \times h(X)$. It is further assumed that the relative degree is maximal, i.e., $q = n$ so that there is no zero dynamics present in the transformed system akin to the continuous case. Define the locally invertible state transformation

$$\zeta = \begin{pmatrix} \zeta_1 \\ \zeta_2 \\ \vdots \\ \zeta_q \end{pmatrix} = \mathcal{T}_d(\boldsymbol{x}) = \begin{pmatrix} h^0(x) \\ h^1(x) \\ \vdots \\ h^{q-1}(x) \end{pmatrix} \tag{3.32}$$

which leads to the normal form

$$\begin{aligned} \zeta_1^+ &= \zeta_2 \\ \zeta_2^+ &= \zeta_3 \\ &\vdots \\ \zeta_{q-1}^+ &= \zeta_r \\ \zeta_q^+ &= h^q(\mathcal{T}_d^{\text{-}1}(\boldsymbol{\zeta}), u) \\ y &= \zeta_1. \end{aligned} \tag{3.33}$$

Using the inverse model (3.31) with $y(k+q) = v$, a linear system of difference equations can be obtained with $\zeta_q^+ = v$. The equations can be represented in the more compact state space notation

$$\begin{aligned} \boldsymbol{\zeta}^+ &= A\boldsymbol{\zeta} + Bv \\ y &= C\boldsymbol{\zeta} \end{aligned} \tag{3.34}$$

for which a static state feedback can be constructed in the form of $v = -K\boldsymbol{\zeta}$ such that the system $\boldsymbol{\zeta}^+ = (A - BK)\boldsymbol{\zeta}$ is stabilized in the origin. Suppose that the inverse model is derived from an imprecise model in the form of (3.25) with $\Phi^*(\boldsymbol{x}, u)$ and $h^*(\boldsymbol{x})$. It is assumed that the imprecise model only differs in its parameters which does not cause any bifurcation in the qualitative behavior of the system. Using the inverse model and combining with (3.33) leads to

$$\zeta_q^+ = h^q(\mathcal{T}_d^{\text{-}1}(\boldsymbol{\zeta}), \Psi^*(\boldsymbol{x}, v)) \tag{3.35}$$

where $\Psi^*(\boldsymbol{x}, v)$ is the imprecise inverse model containing $\Phi^*(\boldsymbol{x}, u)$ and $h^*(\boldsymbol{x})$. The following steps are essentially the same as in the case of the continuous inverse model. One can introduce the requirement

$$\mathcal{E}_d(g) = h^q(\mathcal{T}_d^{-1}(\boldsymbol{\zeta}), \Psi^*(\boldsymbol{x}, g)) - v \tag{3.36}$$

where $g := g(k)$ is an auxiliary control variable. The root g_* is sought again for which $\mathcal{E}_d(g_*) = 0$ is true, which is found by using a deform function, with the property (3.19) that defines a fixed point iteration similarly as in (3.18). The final closed-loop system then takes the form of

$$\begin{aligned} \zeta_1^+ &= \zeta_2 \\ &\vdots \\ \zeta_q^+ &= \mathcal{E}_d(g) + v \\ g^+ &= \mathcal{G}(\mathcal{E}_d(g), g, \boldsymbol{\theta}_g) \\ v &= -K\zeta \\ y &= \zeta_1 \end{aligned} \tag{3.37}$$

One can postulate again that if the deform function has the property (3.24), the controller will stabilize the system in the origin via state feedback.

3.5 Practical considerations and limitations

Several deform functions were introduced throughout the literature, including [R16, R43]. In my thesis, the following deform function was utilized, originating from [R44]:

$$\mathcal{G}(\mathcal{E}(g) - v, g, \boldsymbol{\theta}_g) = (g + K_g)(1 + B_g \tanh(A_g(\mathcal{E}(g) - v))) - K_g, \tag{3.38}$$

where $\boldsymbol{\theta}_g = [A_g,\ B_g,\ K_g]^\intercal$. Due to the nonlinear nature of the controller, there is no standard process to tune the parameters. Nevertheless, a number of heuristics can be used to find the parameters of (3.38), where K_g is set to be a large positive number, A_g is usually a small number with $A_g = 1/(10 \cdot K_g)$ and B_g is either 1 or -1.

The state feedback controller can be replaced by other linear techniques as well. A popular alternative in the RFPT literature is to use the integrating tracking controller [C3]

$$v = \left(\Lambda + \frac{\mathrm{d}}{\mathrm{d}t}\right)^{q+1} \int_0^t e(\tau)\mathrm{d}\tau = 0 \tag{3.39}$$

with the error term $e(t) = y - y^{\text{ref}}$ where y^{ref} is the reference to be tracked and $\Lambda > 0$ is a control parameter, which must be chosen such that the Laplace transform of the time derivative of the error dynamics (3.39) is Hurwitz [R45]. Also, during online iteration when the iteration is close to the fixed point, one can just pass through the previous value to the next one, i.e., if $||g^+ - g|| < \delta$ then $g^+ = g$ (where δ is a sufficiently small number).

There are also a number of limitations associated with the presented framework. I have omitted the derivation of the MIMO case, since the tumor models presented here only have a single input and a single output. Nevertheless, the literature contains MIMO deform functions as well, which were successfully applied in previous works [R46], [Cx1]. For systems with stable zero dynamics, the method can be applied without additional modifications. Nevertheless, in the opposite case, when there are unobservable and unstable modes of the normal form, the method might have to be modified, which should be the focus of later research.

Table 3.1: Identified parameters for the model with 3 states [R37].

s	a	b	c	ED_{50}	K_B	n	w	$x_1(0)$
$S_{0,1}$	0.33	0.12	0.24	$8.89 \cdot 10^{-5}$	0.37	0.12	0.35	0.01
$S_{0,2}$	0.31	0.17	0.3	$9.03 \cdot 10^{-5}$	0.36	0.15	0.34	6.11
$S_{0,3}$	0.31	0.2	0.3	$1.04 \cdot 10^{-4}$	0.34	0.15	0.33	147.58
$S_{0,4}$	0.31	0.18	0.27	$1.33 \cdot 10^{-4}$	0.23	0.17	0.34	51.47
$S_{0,5}$	0.29	0.16	0.31	$8.64 \cdot 10^{-5}$	0.36	0.13	0.34	3.87
$S_{0,6}$	0.3	0.18	0.37	$7.91 \cdot 10^{-5}$	0.37	0.16	0.34	50.75
$S_{0,8}$	0.31	0.17	0.19	$7.79 \cdot 10^{-5}$	0.52	0.13	0.34	11.02
$S_{0,9}$	0.31	0.17	0.16	$8.94 \cdot 10^{-5}$	0.4	0.14	0.34	2.69

Another restriction of the method is that the curse of dimensionality affects the controller design. For systems with large state vectors, the algebraic computation of the inverse model might be intractable. Furthermore, even if the inversion can be performed, it still might be numerically ill-conditioned, leading to poor performance in a closed-loop scenario.

3.6 In silico validation of the RFPT method

Each proposed approach was tested using the tumor growth model with three states, described in (2.2), without impulsive action. At the time of the simulations, eight sets of parameters were identified in [R37] from ten time-series in [R25], shown in Table 3.1. Each subject is denoted by $S_{i,j}$, where i is the experiment index (here $i = 0$, which is consistent with the indexing in Chapter 5.), and j is the identifier for each time-series in the given experiment (here $j \in \{1, \dots, 9\}$ for $i = 0$).

Computation of the coordinate transform (3.6), its inverse transformation, and the terms in the inverse model (3.8) was carried out using Wolfram Mathematica [C1]. In the continuous case, the coordinate transform was computed to be

$$\begin{aligned}
\xi_1 &= x_1 + x_2 \\
\xi_2 &= (a-n)x_1 + nx_1 - wx_2 \\
\xi_3 &= a\left((a-n)x_1 - \frac{bx_1x_3}{ED_{50}+x_3}\right) - w\left(nx_1 - wx_2 + \frac{bx_1x_3}{ED_{50}+x_3}\right),
\end{aligned} \tag{3.40}$$

with the inverse transformation

$$\begin{aligned}
x_1 &= \frac{w\xi_1 + \xi_2}{a+w} \\
x_2 &= \frac{a\xi_1 - \xi_2}{a+w} \\
x_3 &= -\frac{ED_{50}(w\xi_2 - a(w\xi_1+\xi_2) + n(w\xi_1+\xi_2) + \xi_3)}{nw\xi_1 + n\xi_2 + w\xi_2 - a(w\xi_1+\xi_2) + b(w\xi_1+\xi_2) + \xi_3}.
\end{aligned} \tag{3.41}$$

The remaining terms $\alpha(\boldsymbol{\xi})$ and $\beta(\boldsymbol{\xi})$ are

$$\begin{aligned}
\alpha(\boldsymbol{x}) &= x_1\left(a - n - \frac{bx_3}{ED_{50}+x_3}\right)\left(a^2 - an - nw - \frac{b(a+w)x_3}{ED_{50}+x_3}\right) \\
&\quad - w^2\left(nx_1 - wx_2 + \frac{bx_1x_3}{ED_{50}+x_3}\right) - \frac{bED_{50}(a+w)x_1x_3(c(ED_{50}+x_3))}{(ED_{50}+x_3)^3(K_b+x_3)} \\
\beta(\boldsymbol{x}) &= -\frac{bED_{50}(a+w)x_1}{(ED_{50}+x_3)^2},
\end{aligned} \tag{3.42}$$

where they are represented in the original coordinates $\boldsymbol{x} = \mathcal{T}^{\text{-}1}(\boldsymbol{\zeta})$. The corresponding linear controller was

$$v = k_f y^{\text{ref}} - K\boldsymbol{\xi} \tag{3.43}$$

$$k_f = -\frac{1}{C(A - BK)^{\text{-}1}B}, \tag{3.44}$$

where K is the gain, which was determined using pole placement, where the poles are introduced later, since their values differ with each variant. The matrices in (3.44) were computed to be

$$A = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}, \; B = \begin{pmatrix} 0 & 0 & 1 \end{pmatrix}^{\mathsf{T}}, \; C = \begin{pmatrix} 1 & 0 & 0 \end{pmatrix}. \tag{3.45}$$

The closed-loop poles $p_{cl} = [-0.01, -0.02, -0.03]$ were chosen so that the response of the system is a damped curve with no overshoot when uncertainties are not present. Since the original system is bound to be strictly positive, it is required, that the controller should produce an input signal which has no zero crossings and is positive for all times. The previous choice p_{cl} ensures such a response. Furthermore, the reference trajectory was chosen to be $y^{\text{ref}} = 1$.

The deform function used in the simulations was chosen to be (3.38). The initial condition for the fixed-point iteration was computed as $g_0 = v$ for the IO linearized system with no parameter discrepancy. The three control parameters were tuned also assuming no parameter mismatch in the model, from which $B_c = -1$, $K_c = 100$ and $A_c = 0.001$ (shown in (3.38)) resulted in qualitatively the same output signal which is produced by the linear controller. The number of fixed point iterations was chosen to be $M_g = 100$.

The robustness of the algorithms was tested by averaging the model parameters, associated with controllable tumors in Table 3.1, which then were applied to the inverse model. The controllable tumors mean parameters sets, where the tumor volume can be arbitrarily controlled through the input variable. Here the controllable tumors were $S_{0,1}$, $S_{0,2}$, $S_{0,3}$, $S_{0,4}$, $S_{0,5}$, $S_{0,6}$. The average model parameters were then $\boldsymbol{p}^{\text{avg}} = [0.302,\ 0.178,\ 6.1 \cdot 10^{\text{-}7},\ 0.31,\ 9.86 \cdot 10^{\text{-}5},\ 0.33,\ 0.15,\ 0.33]$. The motivation behind this choice is that in a fixed-effect estimation (presented in Chapter 4.1.), a population average is computed, which can be used as an estimate of the true model parameters of the subject in the inverse model.

For the purely continuous case, the **ode45** solver was used to compute g_0 with default settings. Results on the purely continuous RFPT method were then obtained by simulating (3.20) using the **ode15s** stiff solver since the parameter discrepancies resulted in stiff ODEs, for which **ode45** was not able to provide a solution.

Simulation for the different virtual patients for the experimental setup lead to satisfying results in general. The controller could provide setpoint tracking in the cases, where no drug resistance was present, where the only exception was $S_{0,4}$. In this case, the solver encountered a singularity, thus the solution was not available for these control parameters. The problem could be solved by increasing the value of A_c to $A_c = 0.009$, which lead to a satisfying solution. Nevertheless, in all cases, the produced input signal was positive for all time and tracking was achieved. An example result with the parameters $S_{0,2}$ can be seen in Figure 3.1, where the evolution of state variables, and the control variable g is shown in the first and last iterations.

The online iterations were also tested on the same parameter set, for which the corresponding algorithm is presented in Algorithm 1. The model was discretized using $\Delta t = 0.0001$, while the other control parameters remained the same. Results were the same in this setting as well, including the solution issues in $S_{0,4}$. In this case, however, no parameter change resulted in valid solutions. This can be attributed to the Euler integration

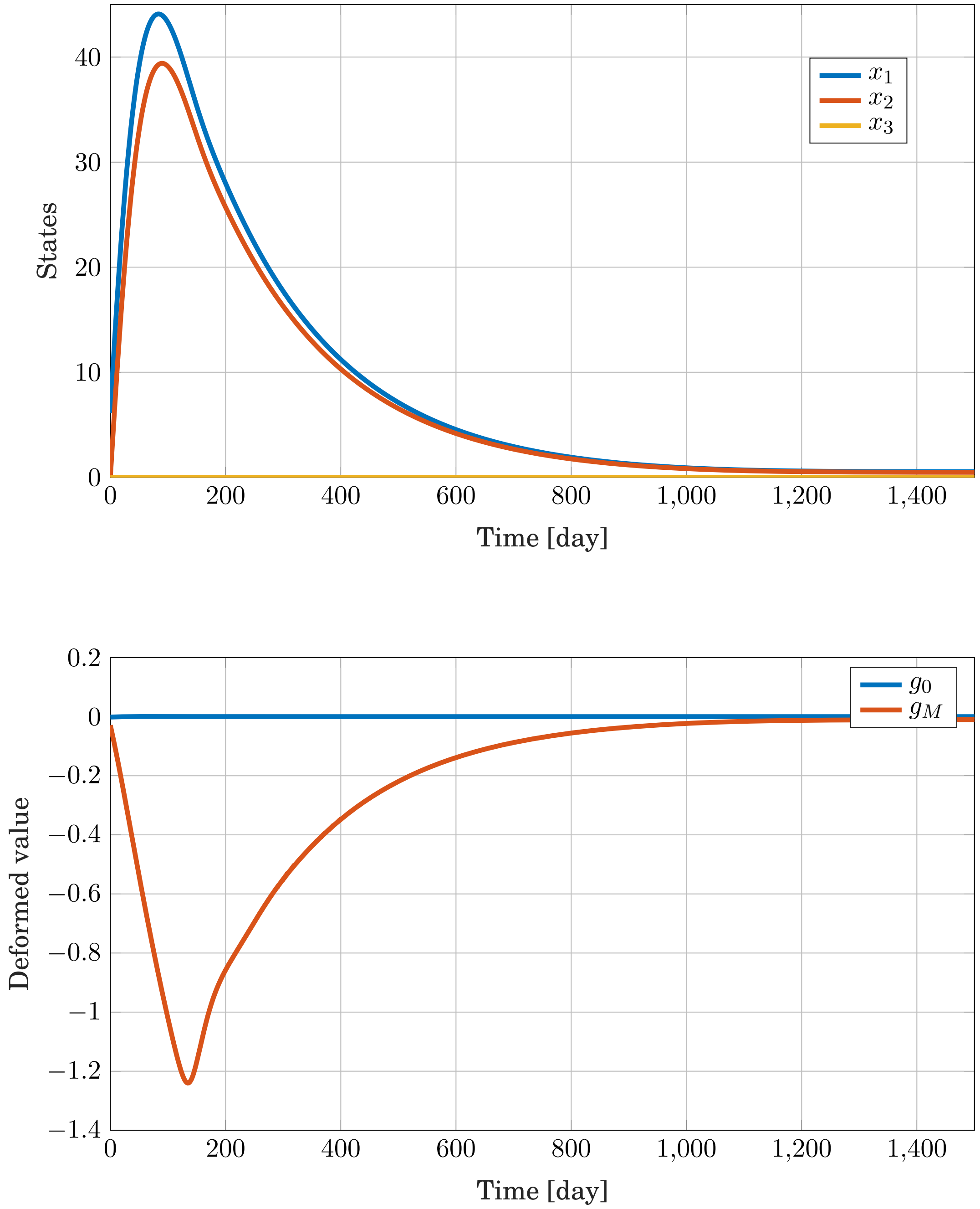


Figure 3.1: Example result for parameters, corresponding to specimen $S_{0,2}$ in the case of the purely continuous variant of the RFPT method. The first diagram shows the evolution of the three state variable. The second plot shows the corresponding deform function, where one can see that the original IO linearized control signal $g_0 = v$ is altered, resulting in g_M.

scheme used in this variant, which resulted in a singularity at time $t = 18.42$. The issue could not be resolved even varying the step size. Nonetheless, in the other cases, the controller produced the positive inputs, and the original IO linearized trajectories were recovered by the deform function. An example solution can be seen in Figure 3.2, for the subject $S_{0,2}$.

Algorithm 1: Real-time RFPT

```
Control parameters: $A_c = 0.009, B_c = -1, K_c = 100$, $p_{cl} = [-0.01, -0.02, -0.03]$, $k_f$ (3.43), $y^{\text{ref}} = 1$, $\Delta t = 0.0001$
System parameters: $A, B, C$ (3.45), $\boldsymbol{x}_0 = [x_1(0), 0, 0]$
$\boldsymbol{p}^{\text{avg}} = [0.302,\ 0.178,\ 6.1 \cdot 10^{-7},\ 0.31,\ 9.86 \cdot 10^{-5},\ 0.33,\ 0.15,\ 0.33]$
Compute $\boldsymbol{\xi}_0$ assuming $\boldsymbol{p}^{\text{avg}}$: $\boldsymbol{\xi}_0 = \mathcal{T}_c(\boldsymbol{x}_0)$ (3.40)
Calculate $g := g_0 = k_f y^{\text{ref}} - K\boldsymbol{\xi}_0$
while Control is active do
    Measure $\boldsymbol{\xi}$ and $\dot{\xi}_r$
    $v = k_f y^{\text{ref}} - K\boldsymbol{\xi}$
    $g^+ = (g + K_g)(1 + B_g \tanh(A_g(\dot{\xi}_3 - v))) - K_g$ (3.38)
    if $|g^+ - g| < \delta$ then
        $g^+ = g$
    $u = \frac{g - \alpha^*(\boldsymbol{\xi})}{\beta^*(\boldsymbol{\xi})}$, using $\boldsymbol{p}^{\text{avg}}$ as the model parameters (3.42)
    Apply the control signal $u$ and let $g := g^+$
```

The last variant was the purely discrete algorithm [C1], for which the corresponding algorithm is presented in Algorithm 2. In discrete-time, the coordinate transform is given as

$$\begin{aligned}
\zeta_1 &= x_1 + x_2 \\
\zeta_2 &= x_1 + a\Delta t x_1 + x_2 - \Delta t w x_2 \\
\zeta_3 &= \frac{1}{ED_{50} + x_3}(ED_{50}((1 + a^2\Delta t^2 + a\Delta t(2 - \Delta t n) - \Delta t^2 n w)x_1 \\
&\quad + (-1 + \Delta t w)^2 x_2) + ((1 + a^2\Delta t^2 - a\Delta t(-2 + b\Delta t + \Delta t n) \\
&\quad - b\Delta t^2 w - \Delta t^2 n w)x_1 + (-1 + \Delta t w)^2 x_2)x_3),
\end{aligned} \tag{3.46}$$

with the corresponding inverse transformation

$$\begin{aligned}
x_1 &= \frac{(-1 + \Delta t w)\zeta_1 + \zeta_2}{\Delta t(a + w)} \\
x_2 &= \frac{\zeta_1 + a\Delta t \zeta_1 - \zeta_2}{\Delta t(a + w)} \\
x_3 &= -\frac{ED_{50}((1 + a\Delta t - \Delta t n)(-1 + \Delta t w)\zeta_1 + (2 + a\Delta t - \Delta t(n + w))\zeta_2 - \zeta_3)}{(1 + a\Delta t - b\Delta t - \Delta t n)(-1 + \Delta t w)\zeta_1 + (2 + a\Delta t - b\Delta t - \Delta t n - \Delta t w)\zeta_2 - \zeta_3}.
\end{aligned} \tag{3.47}$$

The remaining terms, $h^r(\boldsymbol{x}, u)$ (A.1) and $\Psi(\boldsymbol{x}, v)$ (A.2) can be found in the appendix due to formatting considerations.

Here, a different discretization constant $\Delta t = 1$ was used. Note that this value is significantly larger than in the case of the online iteration scheme. This is due to the reason, that the inverse transformations are numerically unstable, when $\Delta t \to 0$. For discrete-time linear systems, the stable poles lie in the unit disk around the origin. To this end, the closed-loop poles were chosen to be $p_{cl} = [0.9,\ 0.8,\ 0.7]$, which results in a damped

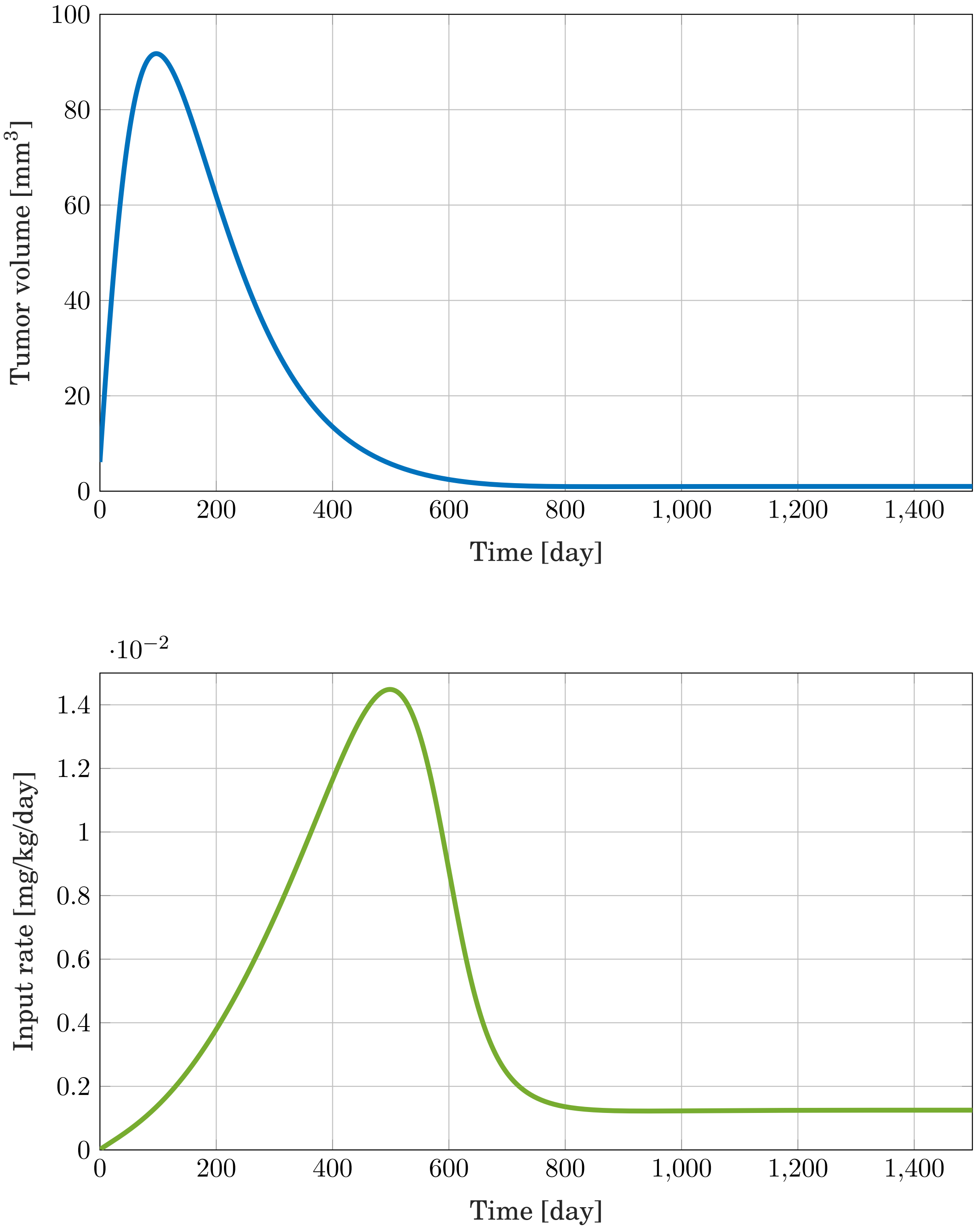


Figure 3.2: Example result for parameters, corresponding to specimen $S_{0,2}$ in the case of the online iteration. This first plot shows the evolution of the tumor volume, and the second plot contains the corresponding input signal.

behavior for tracking, while the parameters of the deform function remained unchanged. The correction term for the linear controller was also modified to

$$k_f = -\frac{1}{C(I - A - BK)^{-1}B}, \tag{3.48}$$

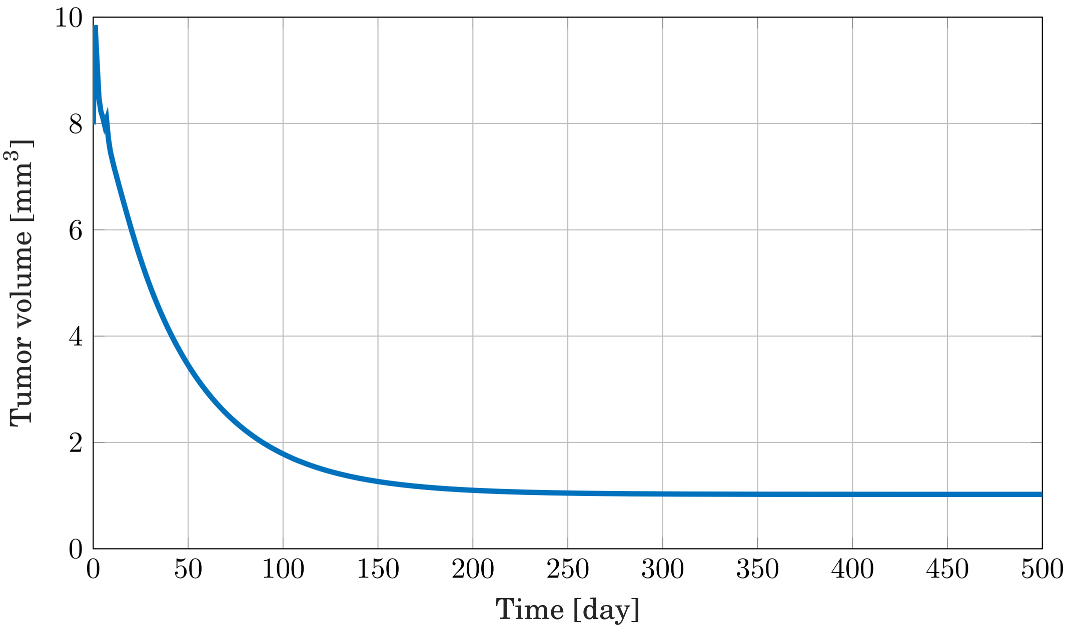
where $I \in \mathbb{R}^{3\times3}$ is the identity matrix [R47]. Simulating the system with the same configuration, i.e., using average model parameters in the inverse model, lead to unsatisfactory results in this case. The controller, could only provide satisfactory results for $S_{0,2}, S_{0,5}$ and $S_{0,6}$, while in the other instances the numerical errors lead to unstable trajectories. In Figure 3.3. the result of the discrete-time RFPT controller is shown for $S_{0,2}$. Note that the input in this case is not strictly positive, hence this solution is not valid for the tumor growth model.

Algorithm 2: Discrete time RFPT

```
Control parameters: $A_c = 0.009, B_c = -1, K_c = 100, p_{cl} = [0.9, 0.8, 0.7]$, $k_f$ (3.43),
  $y^{\text{ref}} = 1$, $\Delta t = 1$
System parameters: $A, B, C$ (3.45), $\boldsymbol{x}_0 = [x_1(0), 0, 0]$
$\boldsymbol{p}^{\text{avg}} = [0.302,\ 0.178,\ 6.1\cdot10^{-7},\ 0.31,\ 9.86\cdot10^{-5},\ 0.33,\ 0.15,\ 0.33]$
Compute $\boldsymbol{\zeta}_0$ assuming $\boldsymbol{p}^{\text{avg}}$: $\boldsymbol{\zeta}_0 = \mathcal{T}_d(\boldsymbol{x}_0)$ (3.46)
Calculate $g := g_0 = k_f y^{\text{ref}} - K\boldsymbol{\zeta}_0$
while Control is active do
    Measure $\boldsymbol{\zeta}$ and $\zeta_r^+$
    $v = k_f y^{\text{ref}} - K\boldsymbol{\zeta}$
    $g^+ = (g + K_g)(1 + B_g \tanh(A_g(\zeta_3^+ - v))) - K_g$
    if $|g^+ - g| < \delta$ then
        $g^+ = g$
    $u = \Psi(\mathcal{T}_d^{-1}(\boldsymbol{\zeta}), g)$, using $\boldsymbol{p}^{\text{avg}}$ as the model parameters (A.2)
    Apply the control signal $u$ and let $g := g^+$
```

The RFPT method performed well in the continuous cases, and lead to robust setpoint tracking in the majority of parameter configurations. However, the discrete-time variant did not prove to be useful in this case due to the inherent structural conditioning of the transformation rules. Additionally, each solution assumes continuous administration which is currently infeasible.

In the following, a different approach using Model Predictive Control and Moving Horizon Estimation is presented which permits the use of impulsive action that models the administration realistically, and can provide the same robust behavior as the RFPT controller.



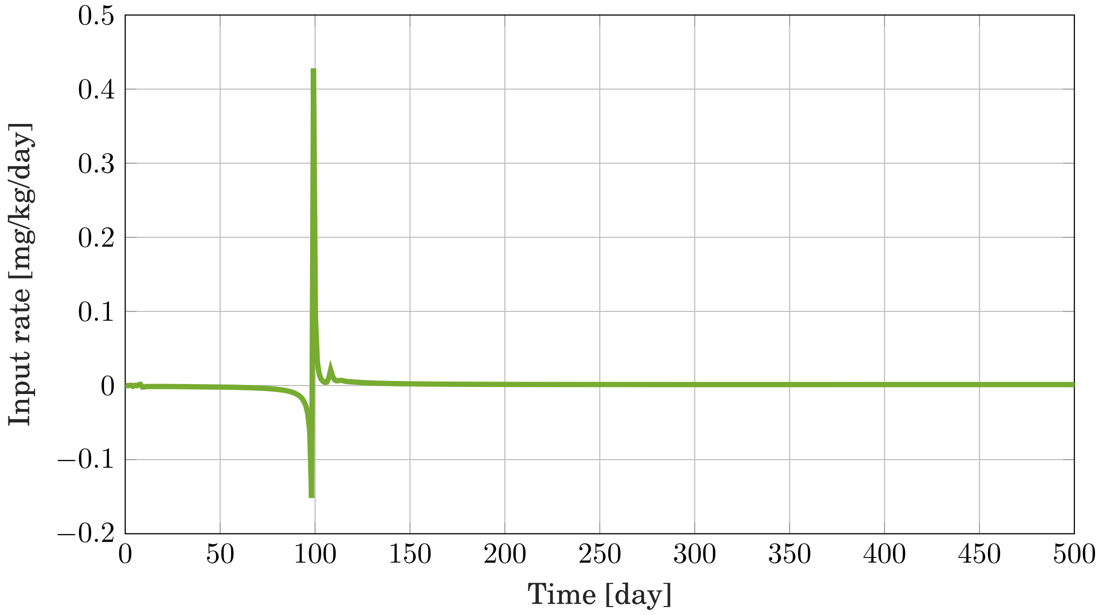


Figure 3.3: Example result for parameters, corresponding to specimen $S_{0,2}$ in the case of the discrete variant. This first plot shows the evolution of the tumor volume, and the second plot contains the corresponding input signal.

4

Chemotherapy optimization

Thesis Group 2: Chemotherapy optimization using NMPC and MHE

Thesis 2
I designed an impulsive NMPC to compute individualized chemotherapy protocols for mice. The controller was augmented with an MHE and the combined robustness of the method was tested in silico using a virtual population that was generated from previous mice experiments.

Publications relevant to the theses: [C8, C7, C5, C6].

Thesis 2.1
I implemented an impulsive NMPC using Direct Multiple Shooting with Dirac delta approximations in the input action. The results were demonstrated in silico, assuming full state observability and no uncertainties. The simulations showed that the algorithm is capable of providing an optimal administration sequence for the tumor growth model.

The results can be seen in Subsection 4.2.5. and 4.2.10.

Thesis 2.2
I developed a purely impulsive NMPC that uses an impulsive differential equation as the prediction model. I also developed a virtual population that was used for the tuning and testing of the proposed algorithm. Furthermore, I showed that numerical errors in the optimization problem can be avoided by introducing transformations on the state variables and the cost function. Numerical results showed that the numerical stability of the proposed scheme is superior to the DMS based implementation.

The results can be seen in Section 4.2.

Thesis 2.3
I have developed a Moving Horizon Estimator that is able to simultaneously estimate the states and the parameters of the underlying tumor growth model. I also presented the tuning process of the estimator. In silico results on previous experimental data indicated that the algorithm is effective in solving the estimation task.

The results can be seen in Section 4.1.

In the following section a full description of the control method applied in the experiments is given. First, the parameter estimation of the model is discussed, followed by the description of the state estimation algorithm, provided by the MHE. This is then followed by the description of the NMPC, where several implementations were examined.

4.1 State and parameter Estimation

In Chapter 2., I introduced two tumor growth models, which can be used to design model-based therapies. In particular, the model of choice contains only three or four state variables that describe the dynamics of tumor growth in conjunction with the pharmacokinetics and pharmacodynamics of the therapeutic regime. While the simplicity of the model is attractive from the perspective of control design, some modification is essential to ensure proper predictive capabilities. In this chapter, I investigate the problem of parameter estimation of the four state model from time-series data and provide modifications that can alleviate technical difficulties of the procedure through the use of the Full Information Estimator (FIE) and the Moving Horizon Estimation (MHE).

4.1.1 Parameter identifiability

Whenever a parameter identification task is present, an identifiability analysis should be conducted before to determine which parameters can be identified. Such an analysis was performed for the model with three state variables in [R48]. The paper used the GenSSI Matlab toolbox [R49] to perform the identifiability analysis, and the AMIGO 2 Matlab toolbox [R50] for sensitivity analysis. The paper showed that for a particular initial condition and input sequence, the parameters a, b, n, and w were the most sensitive, and they all can be structurally locally identified (s.l.i.). This analysis formed a basis for the design of an MHE in the first two experimental validations, described in [R51]. In these two cases, only the parameters a, b, n, and w were identified during the experiments using the MHE, and the rest were assumed to be constant. Despite conducting a prior extensive in silico validation before the in vivo experiments, the MHE in both cases led to significant variations in the estimated parameters. The variation is caused by the s.l.i. property of the parameters, which means that an identification problem (least squares fit on the measured tumor volume data) could have multiple solutions. This implies that even if two consecutive measurements are close to each other, the corresponding identified parameters might have drastically different values, leading to poor performance in control (as can be seen in Chapter 5).

In order to overcome this issue, the goal was to redesign the MHE such that these jumps can not occur. Moreover, I also intended to estimate all model parameters simultaneously, not just the most sensitive ones. The reason to do this was the role of ED_{50} and c in the optimization, where the optimized doses highly depend on the value of these two variables. Additionally, I also anticipated that one of the time-varying identified model parameters should correlate with the tumor volume, lending to some qualitative metric on the level of drug resistance in the mice.

To perform the identifiability analysis, I also used the GenSSI Matlab toolbox. GenSSI uses a generating series of Lie derivatives to determine the identifiability of the nonlinear system [R52]. The algorithm only requires the model and an initial condition to determine the identifiability properties of each parameter. The initial condition was determined to be $\boldsymbol{x}_0 = [200, 0.1, 8, 0]$. This choice is based on the in vivo experimental setup, where an identification dose of 8 [mg/kg] is given to a mouse when the tumor exceeds 200 [mm^3]. For the model with four state variables, the algorithm showed that the model parameters $a, b, n, w, ED_{50}, c, k_1$, and k_2 are all s.l.i. Nevertheless, there are a few considerations that must be taken into account regarding the practical identifiability of the model.

For example, when a mouse receives conventional treatment (described in Section 5), a constant 8 [mg/kg] drug is given at every administration instance. Since there is no variation in the input space in this case, there is no hope that ED_{50} can be adequately identified. This issue was present after the baseline experiment (S_0), described in Section 5.1, where the ED_{50} was computed to be in the region $10^{-4} - 10^{-5}$, meaning that even tiny amount of drug leads to remission, which was not the case in practice.

Additional issues emerge in the case of the identifiability of the model as well. If the input is chosen to be purely impulsive, the analysis becomes unreliable since the vast majority of the existing methods assume the vector field of the model to be continuously differentiable. A few exceptions in the literature work on simple problems, but a reliable solution is still in progress [R53, R54].

4.1.2 Mixed-Effect models

The Stochastic Approximation of Expectation Maximization (SAEM) algorithm [R55] was used to identify the parameters of the model in a mixed-effect manner, as indicated in [R36]. Mixed-effect parameters are suitable in cases where repeated measurements are taken on the same process, with some parameter variability between each measurement. In this work, multiple time series are considered on tumor growth, originating from mice experiments, with slight parametric differences between each experiment due to biological variability. Mixed effect parameters consist of two parts, namely the fixed effects, which describe the population average, and the random effects that pertain to each individual sample [R56]. Formally, for systems described with differential equations, one writes

$$y_{sk} = h(\boldsymbol{x}_s(t_k, \boldsymbol{p}_s)) + \varepsilon_{sk} \tag{4.1}$$

where y_{sk} is the kth observed value of time series s, $\boldsymbol{x}_s \in \mathbb{R}^n$ denotes the states of the corresponding system of subject s, dependent on the model parameters $\boldsymbol{p}_s \in \mathbb{R}^p$, $h : \mathbb{R}^n \to \mathbb{R}$ is the output equation of a nonlinear system, and ε_{sk} is a noise term, often characterized by a Gaussian distribution. Furthermore, the parameter has mixed effects, as

$$\boldsymbol{p}_s = \boldsymbol{\theta} + \boldsymbol{\delta}_s,\ \boldsymbol{\delta}_s \sim \mathcal{N}(0,\ \boldsymbol{\Sigma}) \tag{4.2}$$

where $\boldsymbol{\theta}$ describes the fixed effects, while $\boldsymbol{\delta}_s$ is the corresponding random effect for time-series s, drawn from a normal distribution with covariance matrix $\boldsymbol{\Sigma}$. The objective is then to find both $\boldsymbol{\theta}$ and $\boldsymbol{\Sigma}$, which is performed through maximum likelihood (ML) estimation. The procedure which computes the ML estimation is called the Expectation-Maximization (EM) algorithm, which is an iterative algorithm and has different implementations, from which SAEM is preferred due to its numerical efficiency.

One issue that must be addressed with each iterative algorithm is the proper selection of an initial condition, which has a major impact on the final estimated parameters.

Results in [R36] indicate, that the same issue is present in the identification of the parameters in the tumor growth models. Moreover, it is not possible to include box, or any other sophisticated constraints in the SAEM algorithm, which could lead to physiologically unreasonable state trajectories. In the following sections, I detail a possible remedy to this issue.

4.1.3 Full Information Estimator

Instead of using mixed-effect models, one can utilize nonlinear least-squares (NLS) regression methods to estimate the parameters of the model [C8]. The drawback of such an approach is that it does not account for any inter-patient variability in the model, which consideration can be beneficial for virtual population generation, as will be shown in Chapter . Nevertheless, it can help to find proper initial conditions for the SAEM method, by computing the estimates for each time-series independently from the same initial condition, and then taking the average of the results. In the case of NLS, one only considers fixed parameter vectors and no measurement errors as

$$y_{sk} = h(\boldsymbol{x}_s(t_k, \boldsymbol{p}_s)) \tag{4.3}$$

and formulate a least squares minimization problem for a fixed time series s

$$J(\boldsymbol{p}_s^*) := \min_{\boldsymbol{p}_s} \quad \sum_{k=0}^{M} ||y_{sk} - \hat{y}_{sk}(\boldsymbol{p}_s)||^2 \tag{4.4}$$

where $\hat{y}_{sk}(\boldsymbol{p}_s)$ is the simulated output of the system for parameters $\boldsymbol{p}_s$ at time t_k. Finding the optimal parameter vector $\boldsymbol{p}_s$ is, again, performed by an iterative algorithm, which entails the same problem that is present in the case of the SAEM. In order to circumvent this issue, my approach was that instead of taking each measurement simultaneously, $M-1$ number of NLS problems are solved sequentially by considering only the first k sample for the kth NLS optimization. In its essence, this procedure is the same as the FIE algorithm, described in [R23]. Since this approach entails a significant computational complexity, MHE is often used as a substitute in the literature [R57, R58]. Nonetheless, in the case of biological systems, limitations on the frequency of the measurements permits the use of this approach in a time-sensible manner.

The parameter vector for the kth optimization is denoted by $\boldsymbol{p}_k$. In the case of the FIE, one can also augment the parameter vector with the initial condition of the system as $\boldsymbol{p}_k = [\boldsymbol{p}_k, \boldsymbol{x}(t_0)]^\intercal \in \mathbb{R}^{p+n}$ (with a slight abuse of notation), which will also be the subject of optimization. Two data vectors are defined which contain the measurements of the process and the inputs of the system, in the same manner as in Chapter . The time of measurements are again given as $t_0 < t_1 < \cdots < t_k < \cdots < t_M$ with the associated data vector $\boldsymbol{y}_M = [y(t_0),\ y(t_1),\ \ldots y(t_k) \ldots y(t_M)]$. The inputs of the system do not necessarily coincide with the measurements, hence it is defined on $\tau_0 < \tau_1 < \cdots < \tau_i < \cdots < \tau_N$ with its values contained in the vector $\boldsymbol{u}_N = [u(\tau_0),\ u(\tau_1),\ \ldots u(\tau_i),\ \ldots u(\tau_N)]$. Note that $\tau_N < t_M$, since inputs given after the last measurement has no effect on the regression. One can then use these data vectors to define the estimator cost as

$$J_k(\boldsymbol{p}_k; \boldsymbol{p}_{k-1}, \boldsymbol{y}_k, \boldsymbol{u}_k) = \frac{1}{k} \boldsymbol{\Delta y}_k^\top \boldsymbol{\Delta y}_k + \frac{1}{p+n} \boldsymbol{\Delta p}_k^\top R \boldsymbol{\Delta p}_k, \tag{4.5}$$

where $\boldsymbol{y}_k$ and $\boldsymbol{u}_k$ are the truncated versions of the full data vectors, which contain all the measurement and administration data up to the time instance t_k, respectively. The error term $\boldsymbol{\Delta y}_k$ denotes the normalized error between the measured and simulated tumor volumes up to time t_k, i.e. $\boldsymbol{\Delta y}_k = (\boldsymbol{y}_k - \hat{\boldsymbol{y}}_k)/\bar{y}$. Here, $\hat{\boldsymbol{y}}_k$ contains the simulated tumor volumes at each measurement time instant in a vector, and $\bar{y}$ is a normalization constant. The simulated tumor volumes originate from the solution of the underlying differential equation model, which is computed in the interval $[t_0, t_k]$ with initial condition $\boldsymbol{x}(t_0)$, input sequence $\boldsymbol{u}_k$ and model parameters $\boldsymbol{p}_k$. The second error term $\boldsymbol{\Delta p}_k$ penalizes the parametric deviation between each consecutive estimation problem, which is $\boldsymbol{\Delta p}_k = (\boldsymbol{p}_k - \boldsymbol{p}_{k-1}) \oslash \boldsymbol{p_s}$. Here, $\oslash$ denotes the element-wise division between two vectors and $\boldsymbol{p_s}$ contains scaling constants. The weight matrix $R = \text{diag}(\boldsymbol{p}_c)$ is a diagonal matrix with positive entries, contained in the vector $\boldsymbol{p}_c$, that controls the smoothness between each calculated parameter values, i.e., $\boldsymbol{p}_k$ and $\boldsymbol{p}_{k-1}$. The optimization problems with indices $k \in \{1, \dots, M\}$ are defined as

$$\begin{aligned} J_k(\boldsymbol{p}_k^*) := \quad & \min_{\boldsymbol{p}_k \,\in\, \mathbb{R}^{p+n}} && J_k(\boldsymbol{p}_k; \boldsymbol{p}_{k-1}^*, \boldsymbol{y}_k, \boldsymbol{u}_k) \\ & \text{s.t.} && \boldsymbol{x} \in \mathbb{X}, \\ & && \boldsymbol{p}_k \in \mathbb{P} \end{aligned} \tag{4.6}$$

where the sets $\mathbb{X}$ and $\mathbb{P}$ constraint the states and parameters of the system in a biologically meaningful feasible set. In particular, $\mathbb{X}$ constraints the states to be positive, while the constraints imposed on $\mathbb{P}$ can be found in Table 4.1. The solution of the kth optimization is denoted by $J_k(\boldsymbol{p}_k^*)$, where the argument is the optimal parameter vector. Since the optimizations are recursively defined, one needs an initial parameter vector $\boldsymbol{p}_0$ which can be used to initialize the sequential estimation.

When the last optimization $J_M(\boldsymbol{p}_M)$ is solved, the FIE produces an estimate of the model parameters $\boldsymbol{p}_M^*$, which describes the time-series through the simulation of the underlying model. Since in each consecutive step, the deviance between the model parameters is also penalized, the procedure in which the solution is obtained has better convergence properties in practice compared to regression on the whole time series simultaneously. A loose argument behind this effect is that in simultaneous estimation, the problem space is highly nonlinear due to the consideration of every available measurement. By sequentially performing the NLS on each consecutive datapoint, the subproblems are easier to solve, from which the resulting $\boldsymbol{p}_k^*$ is a close approximation of the next problem $\boldsymbol{p}_{k+1}^*$. As a consequence, the introduction of additional cost and constraints on the parameters between each solution results in robust convergence of the method.

4.1.4 Moving Horizon Estimation

Since the structural complexity of the model considered in this work is relatively low compared to microscopic models, the use of time-varying parameters can enhance its predictive capabilities. Contrary to the principles of FIE, the MHE only considers a fixed number of past measurements, thus alleviating the computational burden of the scheme. Furthermore, the estimation of the initial tumor volumes is omitted, such that the last two elements of $\boldsymbol{p}_k$ is no longer subject of the optimization because of the moving horizon.

Here, the number of previous measurements considered at each step will be denoted by W. At each measurement time instance t_k, two data vectors $\boldsymbol{y}_k^W$ and $\boldsymbol{u}_k^W$ are defined. The vector $\boldsymbol{y}_k^W$ contains the previous W number of measurements, i.e., $\boldsymbol{y}_k^W = [y(t_{k-W}),\ y(t_{k-W+1}),\ \dots, y(t_k)]$, while the corresponding input sequence vector $\boldsymbol{u}_k^W$ contains the applied inputs $u(\tau_i), \tau_i \in [t_{k-W}, t_j]$ in between the endpoints of the window. Using the fixed length data vector $\boldsymbol{y}_k^W$ in (4.5), the cost function has the form

$$J_k^W(\boldsymbol{p}_k; \boldsymbol{p}_{k-1}, \boldsymbol{y}_k^W, \boldsymbol{u}_k^W) = \frac{1}{W} \boldsymbol{\Delta y}_k^\top \boldsymbol{\Delta y}_k + \frac{1}{p} \boldsymbol{\Delta p}_k^\top R \boldsymbol{\Delta p}_k. \tag{4.7}$$

Here, the definition of $\boldsymbol{\Delta p}_k$ remains the same, but the error term $\boldsymbol{\Delta y}_k$ is slightly modified. Since a moving horizon is used, simulating the tumor volumes $\hat{\boldsymbol{y}}_k$ requires modified

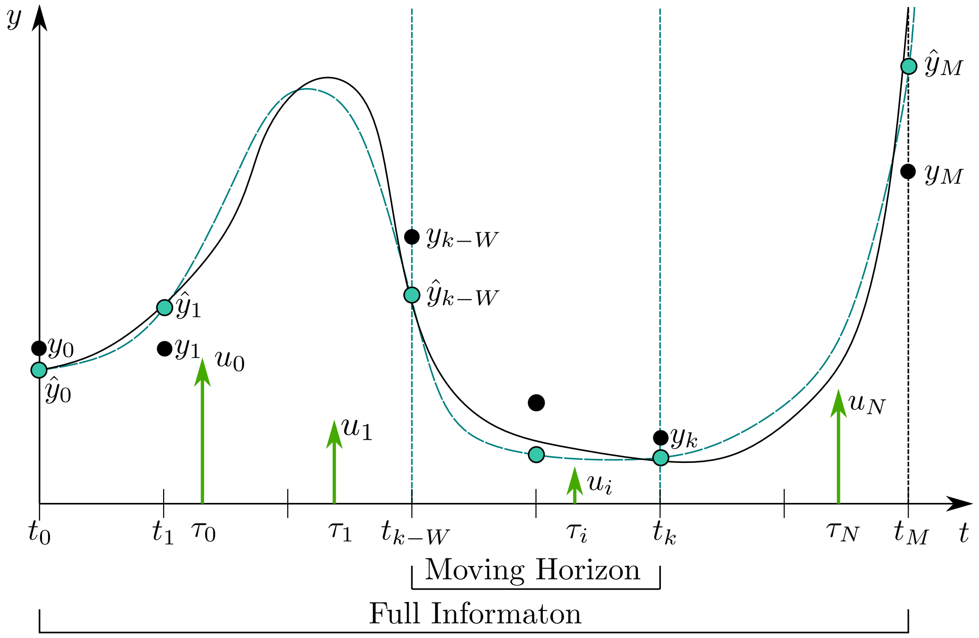


Figure 4.1: Schematic depiction of the estimation process. The black line denotes the true evolution of the tumor, and the black dots represent the measurements on the process. The turquoise line indicates the estimated tumor volume evolution, and the dots represent the estimated measurements at each measurement time instance. The green arrows indicate the dosages, which are not necessarily administered on the same days as the measurement time instances.

input data to the ODE solver. When $k \leq W$ (there are fewer measurements than the length of the window), the initial condition $\boldsymbol{x}(t_0, \boldsymbol{p}_k)$ is used to simulate the system on the interval $[t_0, t_k]$. When $k > W$ (the window start moving), the initial condition is set to be $\boldsymbol{x}(t_{k-W}, \boldsymbol{p}_k) = \boldsymbol{x}(t_{k-W}, \boldsymbol{p}^*_{k-1})$. Using this cost definition, each successive optimization problem can be defined similarly to (4.8) as

$$\begin{aligned} J^W_k(\boldsymbol{p}^*_k) := \min_{\boldsymbol{p}_k \in \mathbb{R}^p} \quad & J^W_k(\boldsymbol{p}_k; \boldsymbol{p}^*_{k-1}, \boldsymbol{y}^W_k, \boldsymbol{u}^W_k) \\ \text{s.t.} \quad & \boldsymbol{x} \in \mathbb{X}, \\ & \boldsymbol{p}_k \in \mathbb{P} \end{aligned} \tag{4.8}$$

A visual depiction of both algorithms can be seen in Figure 4.1. Tuning the parameters of the algorithm, namely the horizon length and the scaling terms, is problem specific and has no general solution. In the following section, I present an approach to tailor the algorithm to the tumor regulation problem and also determine both W and R using the available experimental time series.

4.1.5 Parameter tuning

The parameter tuning of the MHE consists of two steps [C8]. First, the feasible sets $\mathbb{X}$ and $\mathbb{P}$ are determined through appropriate state and parameter constraints. The second step is the computation of the estimator parameters W and R using the FIE. Each time new experiments are conducted, the tuning process can be repeated which eventually yields increased accuracy in the estimation. Therefore, I assume, that N different experimental time-series data is available, where each day, the amount of drug that is given to the mice is recorded, in conjunction with the measured tumor volume values (computed using (2.1)). For each time-series, the vectors $\boldsymbol{u}_{N_s}, \boldsymbol{y}_{M_s}$ represents the input and output data for subject

Table 4.1: Numerical values of the parameters present in both the FIE and the MHE

$\boldsymbol{p}$	$\boldsymbol{p}_0$	$\boldsymbol{p}_0^*$	$\underline{\boldsymbol{p}}$	$\bar{\boldsymbol{p}}$	$\boldsymbol{p}_c$
a	1	0.99	10^{-2}	10	10^{-2}
b	1	1.51	10^{-2}	10	10^{-2}
n	1	0.68	10^{-2}	10	10^{-2}
w	1	0.12	10^{-2}	10	10^{-2}
ED_{50}	1	0.77	10^{-2}	10	10^{-2}
c	1	0.13	10^{-1}	1.5	10^{-2}
k_1	1	4.32	10^{-2}	10	10^{-2}
k_2	1	2.76	10^{-2}	10	10^{-2}
$x_1(t_0)$	$y_0/2$	-	0.5	$\bar{y}$	10^{-2}
$x_2(t_0)$	$y_0/2$	-	0.5	$\bar{y}$	10^{-2}

s. In the first iteration of the parameter tuning, 51 experimental time-series data were available, from which $s \in \{1, \ldots, 51\}$ (corresponding to the baseline, and first experiments $\boldsymbol{S}_0$ and $\boldsymbol{S}_1$, which will be introduced in Chapter 5.). The model used in this section was the tumor growth model with four states, which was introduced in (2.3).

The first step is to identify proper bounds on the states based on the time series. Due to ethical considerations, mice are sacrificed if the tumor volume exceeds a particular value, after which it is considered animal cruelty [R25]. Hence, each experimental time series contains values less than 5000 [mm^3], so it is a suitable choice for the upper bound $\bar{y}$, and the lower bound was chosen to be $\underline{y} = 1$ [mm^3]. The reason behind this choice is that PLD does not entirely eliminate the tumor in the subject, but it is capable of reducing it to such a small extent that the measurement error of the calipers exceeds the true value of the tumor, rendering the obtained values unusable.

The second step is to obtain the upper and lower bounds $\bar{\boldsymbol{p}}$, and $\underline{\boldsymbol{p}}$ on the parameters of the optimization. For the FIE, no prior knowledge is assumed on the parameter values, hence they are set to be uniformly bounded in the region $[0.01, 10]$, which can be seen in Table 4.1, except for the drug clearance parameter. The constraints on c were chosen to be tighter, because values outside these bounds result in immediate, or essentially no depletion of the drug. The initial condition $\boldsymbol{p}_0^*$ of the optimization sequence was selected to be one for all of the model parameters, indicated in Table 4.1. For the initial conditions of the simulation, the initial measured tumor volume was equally divided between $x_1(t_0)$ and $x_2(t_0)$. If the first measurement is zero, the optimization is initialized with $x_1(t_0) = x_2(t_0) = \underline{y}$. This initial condition vector was also used for the parameter scaling vector as $\boldsymbol{p_s} = \boldsymbol{p}_0$.

The last parameters of the FIE that must be found are the weighting terms $\boldsymbol{p}_c$ in the weight matrix R. Since this matrix controls the trade-off between the accuracy of the estimation and the smoothness of the resulting time-varying parameters, it is essential to find the value that balances these two terms. As aforementioned, by penalizing the distance between each consecutive solution $\boldsymbol{p}_k^*$ the convergence of the optimization drastically improves. If one omits this term, the algorithm is not capable of finding an accurate solution to the optimization problem, only if the box constraints are removed, which in turn results in negative parameter values that are physiologically infeasible.

Due to the nonlinear nature of the model, there is no analytic way of determining the optimal estimator parameters that balance the accuracy and parameter deviation, hence requires extensive computational effort to compute their values. As a consequence, a grid search strategy was used by fixing a number of values for $\boldsymbol{p}_c^l$ (second row of Table 4.2.) and defining $R = \text{diag}(\mathbf{1} p_c^l)$, where $\mathbf{1} \in \mathbb{R}^{10}$ is a vector with all of its entries being equal to one.

During the search, the **ode45** routine of MATLAB was employed to simulate the model

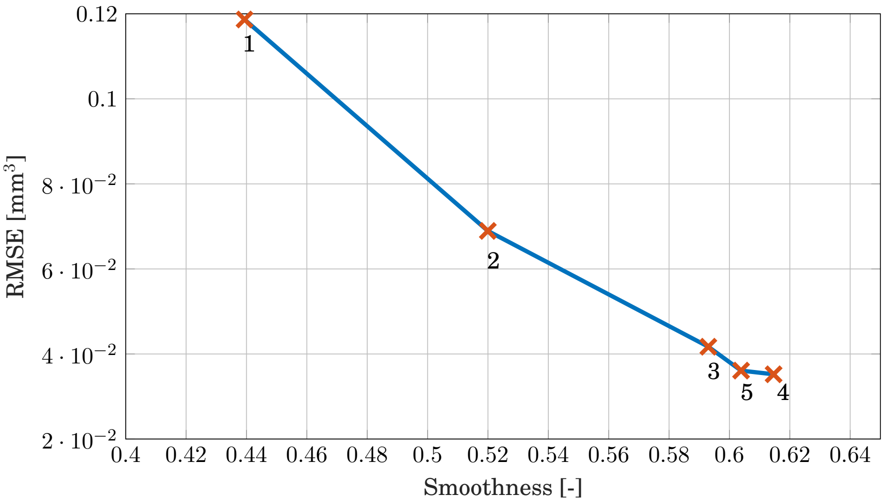


Figure 4.2: Diagram between the overall RMSE and smoothness scores associated with each weight parameter.

(2.3), where the *NonNegative* option was set to be true. This is an important attribute because, for some combination of the parameters, the numerical errors result in negative state values which in turn leads to failure in the optimization. For the optimization, the **fmincon** routine was used to solve problem (4.8). In particular, the *SQP* solver was chosen, where the *MaxFunctionEvaluations* parameter was changed to 15000 and the *FiniteDifferenceType* parameter was set to *central* in order to improve convergence.

At the end of the optimization, two different scores were introduced to measure the accuracy and the smoothness of the solutions, corresponding to each subject for each $\boldsymbol{p}_c^l$. For a fixed time series s, the accuracy was determined using the normalized Root Mean Square Error (RMSE), which is computed using

$$RMSE_s = \sqrt{\frac{1}{M_s}\boldsymbol{\Delta y}_{M_s}^{\mathsf{T}}\boldsymbol{\Delta y}_{M_s}}. \tag{4.9}$$

Again for a fixed s, the matrix $\boldsymbol{P}_{M_s}^* = (\boldsymbol{p}_0^*,\ \boldsymbol{p}_1^*, \dots, \boldsymbol{p}_{M_s}^*)$ is defined, which contains the time-varying parameters computed at each measurement time instance. The *lag one sample autocorrelation* function was used to determine the smoothness of the parameters, associated with $\boldsymbol{P}_{M_s}^*$, which is denoted by $\tilde{\boldsymbol{p}}_a^*$, using the **autocorr** function of MATLAB with default settings. Values closer to one indicate smooth behavior, while values near zero imply sudden changes between the estimations. Because this score has an opposite direction, compared to the RMSE (the higher the score the better the result), the value was transformed as $\tilde{\boldsymbol{p}}^* = -\tilde{\boldsymbol{p}}_a^* + \mathbf{1}$. Finally, an overall score is obtained on the smoothness of the parameters by taking the average on the elements of $\tilde{\boldsymbol{p}}^*$ for the time series s.

By computing these scores for each time series, one can average them to obtain a single score for each possible weight $\boldsymbol{p}_c^l$. Table 4.2 contains the computed scores associated with weights $\boldsymbol{p}_c^l$. Based on the values, one can see that both scores stay relatively constant after 10^{-2}. In addition to that, using smaller weights often results in futile iterations during the estimation, which is the reason why the RMSE of the fourth instance is lower than the fifth, as can be seen in Figure 4.2. Considering these observations, the value $\boldsymbol{p}_c = 10^{-2} \cdot \mathbf{1}$ was chosen, which can also be seen in Table 4.1.

After the search, 51 constant model parameter instances were obtained, each denoted by $\boldsymbol{p}_s^*$. In Figure 4.3., the box plot of the obtained model parameters is shown to visualize their distributions [C8]. In order to improve the initial estimation, the mean of each parameter was taken and used as an initial estimate in the SAEM algorithm. Because

Table 4.2: Smoothness and RMSE scores of different tuning parameters

l	1	2	3	4	5
p_c^l	10^0	10^{-1}	10^{-2}	10^{-3}	10^{-4}
RMSE	0.12	0.07	0.04	0.035	0.036
Smoothness	0.44	0.52	0.59	0.61	0.6

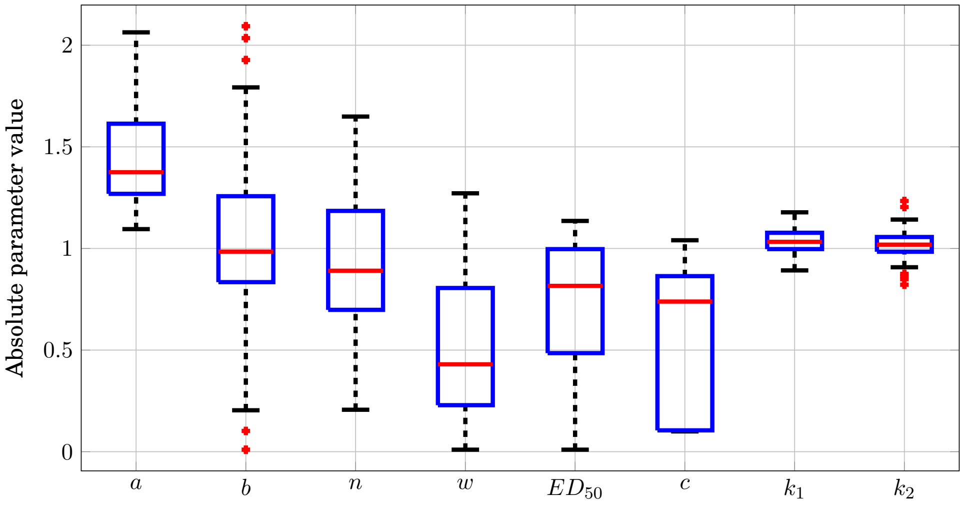


Figure 4.3: Box plot of the obtained parameters on the 51 time series. The visualization of $x_1(t_0)$ and $x_2(t_0)$ was excluded since they will vary significantly due to the various initial measurements contained in the time series. On each box, the central red line indicates the median, and the bottom and top edges of the box indicate the 25th and 75th percentiles, respectively. The whiskers extend to the most extreme data points not considered outliers, and the outliers are plotted individually using the red + marker.

SAEM is a stochastic algorithm, 100 instances of the algorithm were executed, each one initialized from the same mean parameter vector. The computation was carried out by using **nlmefitsa** routine of MATLAB with log transformation on the parameters to constrain them to be positive. Since SAEM does not allow box constraints to be imposed during the identification, those results were filtered where the parameters exceed the region $[\underline{\boldsymbol{p}}, \bar{\boldsymbol{p}}]$. The remaining instances then were ordered based on their RMSE scores, from which the lowest instance was chosen, denoted by $\boldsymbol{p}_0^*$. This value is then used in the MHE as an initial guess for the parameter vector. Because MHE does not estimate the initial states of the model, these values are omitted from Table 4.1.

In order to find a proper value for W, a different approach was taken. Based on biological considerations, the cell homogeneity of the tumor is altered in the presence of a chemotherapeutic agent. In a simplified model, each cell can die, survive or adapt to the molecular configuration of the given drug, leading to mutations in the tumor [R26]. Each successive treatment then entails some form of mutation which eventually leads to drug resistance. Therefore, I assumed that the parameter change of the tumor happens during the clearance of a single dose of 8 [mg/kg], which, according to [R25], is around 5 to 7 days. Based on this consideration, the window length of the MHE was chosen to be $W = 6$.

4.2 Nonlinear Model Predictive Control

Nonlinear Model Predictive Control (NMPC) is an optimal control methodology that has become one of the most popular nonlinear control strategies due to its flexibility [R59]. In classical optimal control, one seeks to solve an infinite-dimensional optimization problem on a finite interval by minimizing a given cost function, subject to the system's dynamics. Since many systems are nonlinear, the optimization problem can not be solved analytically and requires numerical ODE solvers in conjunction with nonlinear optimizers. The resulting optimal control signal is then applied to this finite interval of the system in an open-loop manner. NMPC is based on the same principles, except the optimization is performed each time a new measurement is available. This also entails that the solution interval is also advancing temporally, thus resulting in a closed-loop control scheme. Solving the optimal control problem multiple times in succession requires extensive computational capacities, but with the current advancements in technology and algorithmic design, NMPC has gained widespread use among control engineers in both industry and academy [R60, R61, R62].

In the current chapter, I describe the theoretical background of the NMPC, based on the works of [R24]. While most of the literature introduces NMPC using discrete time processes [R63, R59], the following exposition focuses on implementing continuous systems. First, the general optimal control formulation is presented, followed by the different discretization approaches that can be utilized to approximate the infinite-dimensional problem. The implementation of the controller to the tumor regulation problem will also be presented, with the tuning of the control parameters and in silico investigations on the method's performance.

4.2.1 Optimal Control

In optimal control, the problem is often defined as minimizing some functional on a finite interval that is parametrized by the control input of the system. In technical terms, the optimal control problem (OCP) is given by

$$\begin{aligned} J_T(\boldsymbol{x}^*(t), u^*(t)) = \min_{u(t)} \quad & \int_0^T L(\boldsymbol{x}(t), u(t))\mathrm{d}t + E(\boldsymbol{x}(T)) \\ \text{s.t.} \quad & \boldsymbol{x}(0) - \boldsymbol{x}_0 = 0, \\ & \dot{\boldsymbol{x}}(t) - \boldsymbol{f}(\boldsymbol{x}(t), u(t)) = 0, \\ & \boldsymbol{g}(\boldsymbol{x}(t), u(t)) \geq 0, \\ & \boldsymbol{r}(\boldsymbol{x}(T)) = 0 \end{aligned} \tag{4.10}$$

where $u(t) \in \mathbb{R}$ is a scalar input signal at time t, and $\boldsymbol{x}(t) \in \mathbb{R}^n$ is the state of the system at time t. Furthermore, $L(\boldsymbol{x}(t), u(t))$ is called the stage cost, while $E(\boldsymbol{x}(T))$ is the terminal cost, with T being the terminal time. The first two constraints specify the dynamics of the system with its initial condition $\boldsymbol{x}_0$, $\boldsymbol{g}$ denotes the vector of possible path constraints that the solution must obey (physically reasonable states, or actuator limitations), while the last set of constraints $\boldsymbol{r}$ is the so-called terminal constraints which can be used to stabilize the system.

The solution of the problem $J_T(\boldsymbol{x}^*(t), u^*(t))$ is most often intractable analytically, with the exception of unconstrained linear systems with linear cost, called LQR control. Therefore, the OCP must be discretized on a finite grid of points to solve it numerically. There are three different numerical approaches, which are called direct single shooting (DSS), collocation, and direct multiple shooting (DMS). In this exposition, only DSS and DMS are introduced, which formed the basis of the implemented controllers.

4.2.2 Direct Single Shooting

In the first scenario, only the controls are discretized on a finite grid. Since the goal is the finite parametrization of the infinite problem, one also has to choose the form of the input signal, which is most often a piecewise constant signal. The discrete points on the time grid are denoted by $0 = \tau_0 < \tau_1 < \ldots < \tau_N = T$ and the corresponding constants are $\boldsymbol{d} = [d_0, d_1, \ldots d_N]$, whence $u(t) := u(t, \boldsymbol{d}) = d_i, t \in [\tau_i, \tau_{i+1}]$. Using these discrete variables, the IVP

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t), u(t, \boldsymbol{d})),\ \boldsymbol{x}(0) = \boldsymbol{x}_0,\ t \in [0, T] \tag{4.11}$$

can be solved using numerical ODE solvers, which results in solution vector, where each element in the vector $\boldsymbol{x}_i(\tau_i, d_i)$ is located at τ_i-s. One can also choose a multi-grid approach, where the solution components are given on a finer grid rather than using τ_i-s as basis points. These solution components can be then used to discretize the path and terminal constraints, which transforms the OCP into

$$\begin{aligned} J_T(\boldsymbol{d}) = \min_{\boldsymbol{d}} \quad & \int_0^T L(\boldsymbol{x}(t, \boldsymbol{d}), \boldsymbol{d})\mathrm{d}t + E(\boldsymbol{x}(T, d_N)) \\ \text{s.t.} \quad & \boldsymbol{g}(\boldsymbol{x}(\tau_i), d_i) \geq 0, \\ & \boldsymbol{r}(\boldsymbol{x}(T)) = 0 \end{aligned} \tag{4.12}$$

where the integral is also approximated (using the trapezoidal rule for example) on the discrete points corresponding to each solution component $\boldsymbol{x}_i(\tau_i, d_i)$. DSS is the simplest approximation to the OCP, hence it is often the first choice to implement the optimization.

4.2.3 Direct Multiple Shooting

In the DMS variant, the controls and the states are discretized simultaneously. The discretization of the controls remains the same, and the dynamics of the system is defined piecewise as

$$\dot{\boldsymbol{x}}_i(t) = \boldsymbol{f}(\boldsymbol{x}_i(t), u(t, d_i)),\ \boldsymbol{x}_i(t) = \boldsymbol{s}_i,\ t \in [\tau_i, \tau_{i+1}], \tag{4.13}$$

where each $\boldsymbol{s}_i$ is an artificial initial value that will also be the subject of the optimization. Each piece then numerically solved that result in the trajectory pieces $\boldsymbol{x}(t, \boldsymbol{s}_i, d_i)$. In order to obtain continuous trajectories on the optimization interval, the continuity constraints $\boldsymbol{s}_{i+1} - \boldsymbol{x}_i(\tau_{i+1}, \boldsymbol{s}_i, d_i) = 0$ are imposed during the optimization. For each trajectory piece, the associated cost integral

$$\ell_i(\boldsymbol{s}_i, d_i) = \int_{t_i}^{t_{i+1}} L(\boldsymbol{x}(t, \boldsymbol{s}_i, d_i), d_i)\mathrm{d}t, \tag{4.14}$$

is also approximated. Since the optimization is now dependent on two vectors, namely $\boldsymbol{s} := [\boldsymbol{s}_0,\ \boldsymbol{s}_1,\ \ldots \boldsymbol{s}_N]$ and $\boldsymbol{d}$, one should combine them into a single vector which is compatible with general nonlinear optimizers as $\boldsymbol{w} := [\boldsymbol{s}_0^\mathsf{T}, d_0, \boldsymbol{s}_1^\mathsf{T}, d_1, \ldots, \boldsymbol{s}_N^\mathsf{T}]$. In summary, the OCP is now formulated as

$$\begin{aligned} J_N(\boldsymbol{w}^*) = \min_{\boldsymbol{w}} \quad & \sum_{i=0}^{N-1} \ell_i(\boldsymbol{s}_i, d_i) + E(\boldsymbol{s}_N, d_N) \\ \text{s.t.} \quad & \boldsymbol{s}_0 - \boldsymbol{x}_0 = 0, \\ & \boldsymbol{s}_{i+1} - \boldsymbol{x}_i(\tau_{i+1}, \boldsymbol{s}_i, d_i) = 0, i = 0, \ldots, N-1, \\ & \boldsymbol{g}(\boldsymbol{s}_i, d_i) \geq 0, \qquad\qquad i = 0, \ldots, N, \\ & \boldsymbol{r}(\boldsymbol{s}_N) = 0, \end{aligned} \tag{4.15}$$

where N is the horizon length and $T = N(\tau_{i+1} - \tau_i)$. DMS has a number of practical advantages as opposed to DSS, including faster local convergence and parallelizability of the ODE solution due to the artificial initial values.

4.2.4 Nonlinear Model Predictive Control

Computing the solution of the OCP (4.10) results in a sequence of optimal control signals that can be applied to the system in an open-loop manner. A closed-loop control scheme can be obtained, by measuring the states of the system at each t_k, computing the state estimates at each τ_i, and solving the problem at these instances. This strategy is called the NMPC and has found numerous applications in a wide range of systems, aforementioned in the introduction.

In order to implement the approach online, one has to measure the full-state x of the system at each control instance. However, this is practically infeasible, due to the limited number of observables during the experiment. For instance, one can only measure the complete tumor volume but has no information on the distribution between the living and necrotic parts. This can be alleviated by using a state estimator, as was presented previously in Chapter 4.1. in the form of the MHE.

Consider the solution of the problem (4.8) at time t_k denoted by $\boldsymbol{p}_k^*$. For each $\boldsymbol{p}_k^*$, the differential equations are simulated in the interval $[t_{k-W}, t_k]$ (or from $[0, t_k]$ as described in 4.1.4), resulting in a state estimate $\tilde{\boldsymbol{x}}(t_k)$ at the end of the simulation interval. If the measurement and the input time instances do not coincide, and the input time is after a measurement instance, one can instead simulate the system up to time τ_i with the same model parameters, resulting in the estimate $\tilde{\boldsymbol{x}}(\tau_i)$. These modifications change the OCP as

$$\begin{aligned} J_T(\boldsymbol{x}^*(t), u^*(t), \tau_i) = \min_{u(t)} \quad & \int_{\tau_i}^{\tau_i+T} L(\boldsymbol{x}(t), u(t))\mathrm{d}t + E(\boldsymbol{x}(\tau_i + T)) \\ \text{s.t.} \quad & \boldsymbol{x}(\tau_i) - \tilde{\boldsymbol{x}}(\tau_i) = 0, \\ & \dot{\boldsymbol{x}}(t) - \boldsymbol{f}(\boldsymbol{x}(t), u(t)) = 0, \\ & \boldsymbol{g}(\boldsymbol{x}(t), u(t)) \geq 0, \\ & \boldsymbol{r}(\boldsymbol{x}(t)) = 0. \end{aligned} \tag{4.16}$$

The problem can be solved by either implementing DSS or DMS, as shown in the previous sections. The first element of the control signal $u(\tau_i)$ is then applied to the system, and the procedure repeats at the next control time instance τ_{i+1}. Figure 4.4. visualizes the approach. When the problem is solved numerically, the optimal solution $u^*(t)$ can be passed as an initial condition to the next optimization instant at time τ_{i+1}, which is called the hot-start of the NMPC. This improves the convergence speed of the algorithm, thus making it suitable for real-time applications.

4.2.5 First implementation

In the first implementation of the algorithm, the DMS approach was utilized due to the expected increase in computational speed compared to the DSS [C5]. The initial model in (4.13) was (2.2), containing three state variables. Moreover, the input was represented as the Dirac approximation (2.5) where a single element is given as

$$u_i = u(t, \tau_i, d_i) = \begin{cases} \dfrac{d_i}{2\varepsilon}\left(1 + \cos\left(\dfrac{\pi(t - \xi_i)}{\varepsilon}\right)\right), & \tau_i \leq t \leq \tau_i + 2\varepsilon \\ 0, & \tau_i + 2\varepsilon < t < \tau_{i+1}. \end{cases} \tag{4.17}$$

The stage cost penalizes the tumor volume and the input rate as

$$\ell(\boldsymbol{s}_i, d_i) = \int_{\tau_i}^{\tau_{i+1}} \left(\frac{y(\boldsymbol{s}_i) - y_{\text{ref}}}{y_0}\right)^2 + r\left(\frac{u_i}{\bar{u}}\right)^2 \mathrm{d}t \tag{4.18}$$

where $y(\boldsymbol{s}_i)$ is the simulated output of the system on the interval $[\tau_i, \tau_{i+1}]$, associated with the artificial initial value $\boldsymbol{s}_i$, d_i is the input dose, and r is a control parameter. Moreover, y_0

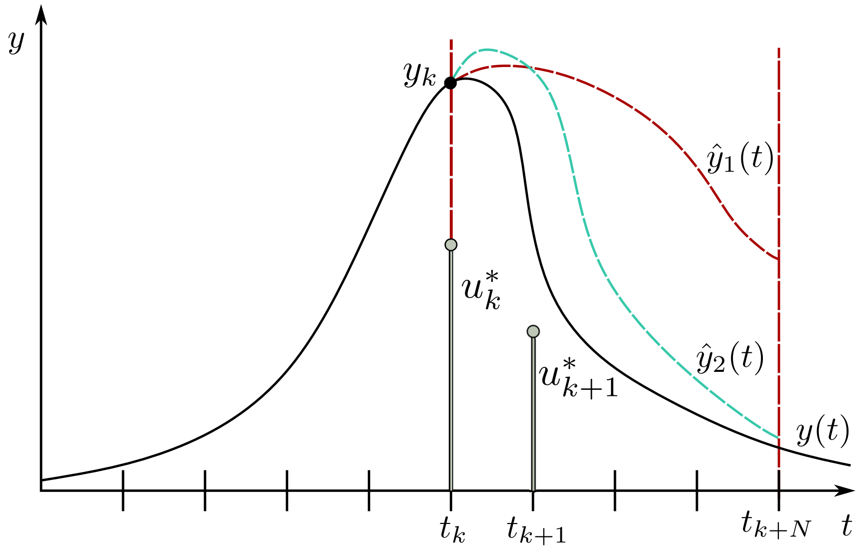


Figure 4.4: Visualization of the NMPC. The red and turquoise lines indicating different predictions concerning different administration protocols during the optimization.

is the measured tumor volume at the beginning of the treatment, $y_{\text{ref}} = 1$ is a nonzero reference tumor volume and $\bar{u} = 480$ is the maximum of (4.17) for $d_i = 8$ and $\varepsilon = (15/86400)/2$. Both scaling terms were introduced to the end of bringing both objectives to the same numerical domain, which can improve the conditioning of the optimization. The optimization problem at time τ_i is then defined as

$$\begin{aligned} J_N(\boldsymbol{w}^*, \tau_i) = \min_{\boldsymbol{w}} \quad & \sum_{l=0}^{N-1} \ell(\boldsymbol{s}_{i+l}, d_{i+l}) \\ \text{s.t.} \quad & \boldsymbol{s}_0 - \boldsymbol{x}_0 = 0, \\ & \boldsymbol{s}_{i+l+1} - \boldsymbol{x}_{i+l}(\tau_{i+l+1}, \boldsymbol{s}_{i+l}, d_{i+l}) = 0, \, l = 0, \dots, N-1, \\ & d_{i+l} \in [0, 8] \qquad\qquad l = 0, \dots, N-1, \\ & \boldsymbol{s}_{i+l} \in \mathbb{R}^{n+} \qquad\qquad l = 0, \dots, N-1, \end{aligned} \tag{4.19}$$

where the notation $\mathbb{R}^{n+}$ means that each element of the state vector is a positive real number. The initial condition for $\boldsymbol{w}$ at the first optimization instance contains the terms $\boldsymbol{s}_{i+l} = [y_0, \ 0, \ 0]$ and $d_{i+l} = 8$. The problem (4.19) was solved using the **fmincon** routine of MATLAB with the options *Algorithm: sqp*, *FiniteDifferenceType: central* and *MaxFunctionEvaluations: 15000*. The integral in (4.18). was approximated using the **trapz** trapezoidal integration method [C6].

4.2.6 Second implementation

Adjustments were made in the second version of the controller on the basis of a number of observations from the first experiment [C7]. The first issue during the experiment was that in some instances, the controller parameters had to be manually changed in order to avoid doses that were too large and might lead to cumulative toxicity in the subjects. Also, for some parameter combinations provided by the MHE, numerical errors emerged which corrupted the resulting doses. In the second generation of the algorithm, a systematic tuning procedure was then introduced to eliminate the numerical errors, with an additional constraint on the cumulative toxicity. The DMS implementation was also replaced since it did not provide the expected computational benefit compared to the DSS. This can be attributed to the fact, that the underlying differential equations contain only a few state variables and are fast to solve using adaptive ODE solvers. By parallelizing the solution, the computational overhead in the implementation is larger than just solely simulating the system. The optimization solver was also changed to a derivative-free method for which improved numerical conditioning was expected. Moreover, the input action was changed to the purely impulsive variant (4.26), since the original formulation turned out to be more computationally demanding without fostering the stability of convergence of the optimization. These changes lead to the reformulated stage cost

$$\ell(y_i, d_i) = \int_{\tau_i}^{\tau_{i+1}} \left(\frac{y(d_i) - y_{\text{ref}}}{y_0} \right)^2 \mathrm{d}t + r \left(\frac{d_i}{\bar{d}} \right)^2, \tag{4.20}$$

where ℓ is similarly defined as in (4.18), except the simulated output depends now on the input sequence only, rather than the artificial initial conditions, which is emphasized by the notation $y(d_i)$. Furthermore, the input effort is no longer integrated, which resulted in control parameters with significantly smaller values as in the previous case. Using the DSS approach, the optimization problem is now given as

$$\begin{aligned} J_N(\boldsymbol{d}; \tau_i) := \min_{\boldsymbol{d} \in \mathbb{R}^N} \quad & \sum_{l=0}^{N-1} \ell(y_{i+l}, d_{i+l}) \\ \text{s.t.} \quad & \boldsymbol{d} \ \in [\boldsymbol{0}, \bar{\boldsymbol{d}}], \\ & \boldsymbol{d^c} \in [\boldsymbol{0}, \bar{\boldsymbol{d}}^{\boldsymbol{c}}], \end{aligned} \tag{4.21}$$

where, $\boldsymbol{d} = [d_i, d_{i+1}, \dots, d_{i+N-1}]$ is the optimal input sequence where each element is constrained to lie in the interval $0 \leq d_{i+l} \leq \bar{d}$, denoted by $\bar{\boldsymbol{d}} = \bar{d}\boldsymbol{1}$ (where $\boldsymbol{1}$ is an N dimensional column vector with all of its entries being equal to 1), and the second group of constraints $0 \leq d^c_{i+l} \leq \bar{d}^c$ denote the cumulative dosage associated with each d_{i+l}, from which $\boldsymbol{d^c} = \bar{d}^c\boldsymbol{1}$. The cumulative dosage d^c_{i+l} is defined here as the sum of doses in the past 10 days,

$$d^c_{i+l} = \sum_J d_j, \ J := \{\, j \mid t_{i+l} - 10 \leq t_j \leq t_{i+l} \,\}. \tag{4.22}$$

According to [R25] the maximal tolerable dose (MTD) of PLD in mice is 8 [mg/kg] which could be repeated in every 10 days without triggering an irreversible weight loss, leading to a threshold of 16 [mg/kg]. To define a safe cumulated dose threshold in the constraint, the maximum given PLD of 16 [mg/kg] was lowered to $\bar{d}^c_{i+l} = 14$ [mg/kg] in 10 days to minimize further the possibility of severe systemic toxicity. The upper bound for each dose was also lowered to $\bar{d}_{i+l} = 6$ [mg/kg] for the same reason. Note that the positivity constraint on the states were omitted here. In the case of the DMS, the artificial initial conditions are also optimized, which can lead to negative state values during the optimization, hence the inclusion of the positivity constraint is essential. By contrast, the DSS approach only simulates the differential equations from positive initial conditions, resulting in a positive trajectory in every instance.

In the first implementation, three days of resting time were assumed between each consecutive treatment, which resulted in fixed prediction intervals in the horizon. Nevertheless, during the experiments, this interval changes, because the administrations take place on Mondays and Thursdays or Tuesdays and Fridays. In practice, this means that the intervals of the optimization change depending on the day on which the optimization takes place, such that

$$\begin{aligned} &(\tau_l, \tau_l + \Delta\tau], \ \Delta\tau = \begin{cases} 3, & \text{if } \tau_l \text{ is Monday or Tuesday} \\ 4, & \text{if } \tau_l \text{ is Thursday or Friday} \end{cases} \\ &\tau_{l+1} = \tau_l + \Delta\tau, \\ &\tau_0 = \tau_i, l \in \{0, \dots, N-1\} \subset \mathbb{N}_0 \end{aligned} \tag{4.23}$$

with an optimization variable d_i at each τ_l [C7]. The optimization problem (4.21) was solved using **fmincon** with the option *Algorithm: patternsearch*. This is a derivative-free method, for which better convergence was expected since I originally assumed that the convergence issues arise from the finite difference calculations during the optimization. Nevertheless, as the second experiment showed, this hypothesis was not true, and additional modifications had to be implemented, which will be shown in the next section.

4.2.7 Third implementation

Identifying the source of numerical errors during the computations was not trivial, and several modifications had to be implemented in order to completely eradicate their presence. In the last iteration of the design, the cost function was altered and additional scaling factors were introduced to this end. One source of such errors could be attributed to the different scales on which the state variables evolve, thus the state variables were rescaled as

$$\mu_1 = \frac{x_1}{x_{1c}},\ \mu_2 = \frac{x_2}{x_{2c}},\ \mu_3 = \frac{x_3}{x_{3c}},\ \mu_4 = \frac{x_4}{x_{4c}},\ \mu_i^u = \frac{u_i}{x_{3c}} \tag{4.24}$$

which lead to the scaled system model

$$\begin{aligned} \dot{\mu}_1 &= (a-n)\mu_1 - b\frac{\mu_1\mu_3}{ED_{50}+\mu_3 x_{3c}} \\ \dot{\mu}_2 &= n\mu_1\frac{x_{1c}}{x_{2c}} + b\frac{x_{1c}}{x_{2c}}\frac{\mu_1\mu_3 x_{3c}}{ED_{50}+\mu_3 x_{3c}} - w\mu_2 \\ \dot{\mu}_3 &= -(c+k_1)\mu_3 + \frac{x_{4c}}{x_{3c}}k_2\mu_4 \\ \dot{\mu}_4 &= \frac{x_{3c}}{x_{4c}}k_1\mu_3 - k_2\mu_4 \end{aligned} \tag{4.25}$$

with the corresponding input rule

$$\boldsymbol{\mu}(t_i^+) = \boldsymbol{\mu}(t_i^-) + (0\ 0\ 1\ 0)^\top \mu_i^u, \tag{4.26}$$

where $x_{1c}, x_{2c}, x_{3c}, x_{4c}$ are scaling constants. The particular values of the scaling parameters were determined to be $x_{1c} = 1,\ x_{2c} = 1\ x_{3c} = ED_{50}$ and $x_{4c} = ED_{50}$. The reason behind this choice is that for small values of ED_{50} in the model, the term

$$b\frac{x_1 x_3}{ED_{50}+x_3} \tag{4.27}$$

has a sharp characteristic when x_3 is varied. Since the optimized dose has a direct effect on x_3, the gradient computed with finite differences can take up high values, leading to no convergence in the optimization. By scaling x_3 to the same region as ED_{50} the problem can be efficiently tackled.

Technically, the scaled model only plays role in the prediction part. Before **fmincon** is called, the initial guess for the optimized doses is scaled by the value of ED_{50}, together with the initial condition of the simulation, provided by the MHE. After the optimization is performed, the resulting optimal vector of doses is then scaled back to their original values, which are then applied to the subjects.

In addition, the cost function was also changed to

$$\ell(y_i, d_i) = \left|\frac{y_i - y_{\text{ref}}}{\bar{y}}\right| + r\frac{d_i}{\bar{d}}, \tag{4.28}$$

where y_i is the simulated tumor volume from the transformed model (4.25) at time τ_i, and $\bar{y}$ is the maximum value that the tumor can attain. The quadratic penalty was replaced with the absolute value function, which also improved the stability of the optimization procedure. The scaling term $\bar{y}$ was chosen to be $5000[\text{mm}^3]$, which is the maximal volume that the tumor can attain before the subject is sacrificed. Initially, it was assumed, that this maximum value is unknown a priori, and the scaling term was set to be equal to the tumor volume at the beginning of the algorithmic therapy. Nonetheless, the performance of numerical optimizers increases if the terms in the cost function are between the interval $[0, 1]$, which justifies the use of the upper bound [R64]. These changes lead to the final optimization problem, which had the same structure as the second variant:

$$\begin{aligned} J_N(\boldsymbol{d}; \tau_i) := \min_{\boldsymbol{d} \in \mathbb{R}^N} \quad & \sum_{l=0}^{N-1} \ell(y_{i+l}, d_{i+l}) \\ \text{s.t.} \quad & \boldsymbol{d} \ \in [\boldsymbol{0}, \bar{\boldsymbol{d}}], \\ & \boldsymbol{d}^{\boldsymbol{c}} \in [\boldsymbol{0}, \bar{\boldsymbol{d}}^{\boldsymbol{c}}]. \end{aligned} \tag{4.29}$$

4.2.8 Virtual population generation

For each version of the controller, a different tuning procedure was used. The first iteration of the design was tuned empirically in silico, where the goal was to vary the control parameters until the first generated dose was smaller than the maximum admissible MTD. In order to automate the process and create a systematic tuning procedure, a virtual population was generated using the SAEM method for the second and third variants [C7]. Since SAEM uses a mixed-effect representation of the model parameters, it is able to compute both the population average and the corresponding covariance matrix. These two values then can be used to generate an arbitrary number of parameter sets, which form the basis of the virtual population. Additional constraints on these model parameters can also be imposed so that the resulting values represent tumors that behave similarly to their real counterparts.

At the time of the second implementation of the controller, only the 10 experimental time-series were available from [R25]. Instead of fitting the parameters of the model directly to each time series, they were cut into multiple intervals, each containing one remission relapse cycle. This means that each cycle begins with tumor growth which is then subject to injection treatment and then followed by a shrink to a minimum value. An example can be seen in Figure 4.5, where the grey vertical lines are the cuts. Resistant artifacts were also excluded, where the injection does not lead to remission, because these fits result in most likely uncontrollable parameter configurations. The primary reason behind cutting the time series is to obtain additional samples on which the identification can be performed. Since the model parameters are assumed to be time-varying, one can think of each cut to be a virtually different subject with its own constant model parameters. Cutting the times series in this way resulted in 21 time series. At the time of the implementation of the second controller, the FIE had not been implemented, and hence, the MATLAB routine **sbiofit** was used with the *scattersearch* option to compute the initial least squares parameter estimates, with the same initial parameter vector and bound constraints that can be seen previously in Table 4.1. These obtained values and also their average was used as an initial condition in separate instances of the SAEM method. At the end of the identification, the parameter resulting in the lowest RMSE score was chosen to determine the fixed-effects and covariance matrix. The estimated, log-transformed fixed effects are contained in Table 4.3 with their standard errors and random effect coefficients. In order to obtain physiologically sensible state trajectories, the deviation of c and k_1 was penalized in **sbiofitmixed** by setting the *Cov0* parameter to diag($[0.1, 0.1, 0.1, 0.1, 0.1, 10^{-6}, 10^{-6}, 0.1, 0.1]$).

According to the mixed-effect principle, one parameter sample is from the normal distribution $\theta_i^* \sim \mathcal{N}(\theta^*, \Sigma^*)$, where θ_i^* is the generated parameter set, θ^* is the estimated

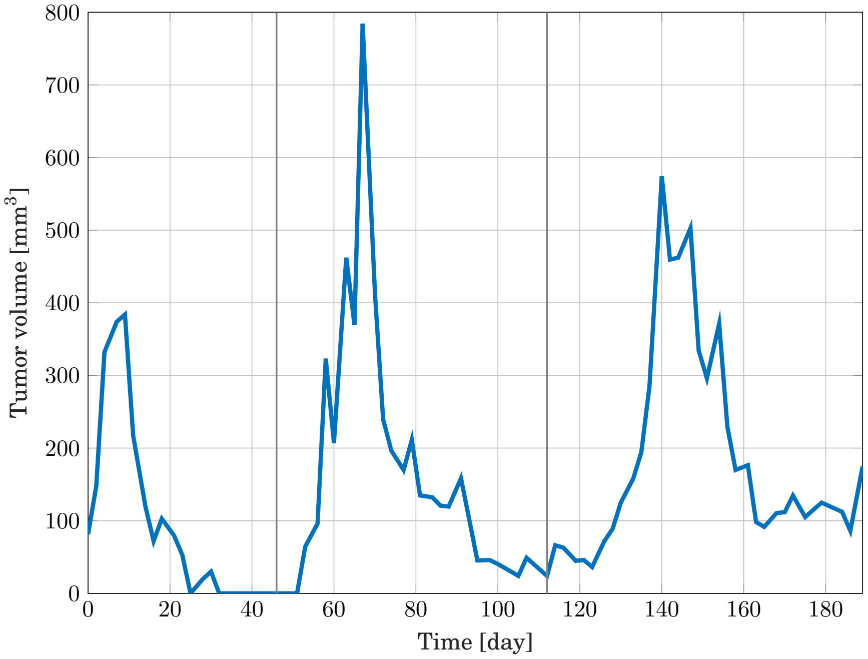


Figure 4.5: An example cut of the in vivo time series $S_{0,6}$. Between each grey line a single growth-shrink cycle is contained which is assumed to be independent from the rest of the time series.

fixed effects in Table 4.3., and Σ^* is the random effect covariance matrix, which is a diagonal matrix with elements from the last column in Table 4.3. Because the values in Table 4.3 are log-transformed to obey the positivity constraint in the parameters, one must back-transform the calculated parameters θ_i^*-s to obtain their values in the original parameter space such that $\theta_i = \exp(\theta_i^*)$. For each sample, a corresponding initial value $\boldsymbol{x}(0) = [x_{10}, 0, 0, 0]$ was drawn from the uniform distribution $x_{10} \sim \mathcal{U}(l_x, u_x)$, where the parameters $l_x = 0, u_x = 1383$ corresponds to the smallest and largest tumor volume where the first dose was applied in the remission-relapse cycle. Statistics were further generated about the cycles which were used to impose certain conditions during the generation such that the generated tumors mimic real dynamics. The first statistics was the mean duration of the cycles $\bar{t}_r = 26.08$ [day] with their standard deviation $\sigma_r = 11.13$ [day]. The second was the mean value of the difference between the peak tumor volume and the starting vol-

Table 4.3: Log-transformed parameter values of the identification for the cut time series from experiment S_0

Parameter	Fixed effect	Back transformed	Standard errors	Random effect covariance
a	-0.84	0.43	0.21	0.087
b	-0.2	0.82	0.17	0.251
n	-2.03	0.13	0.65	0.133
w	-2.44	0.09	0.17	0.417
ED_{50}	-6.71	0.0012	32.7	0.008
c	0.46	$1,58$	3.19	$5.9 \cdot 10^{-8}$
$k1$	1.46	4.3	58.8	$2.38 \cdot 10^{-8}$
$k2$	1.42	4.14	49.96	0.228
x_0	4.08	59.14	0.35	1.78

Table 4.4: Log-transformed parameter values of the identification for 51 subjects, originating from experiments S_0 and S_1.

Parameter	Fixed effect	Back transformed	Standard errors	Random effect covariance
a	0.26	1.3	0.18	0.013
b	0.03	1.03	0.21	0.078
n	0.01	1.01	0.23	0.018
w	-1.89	0.15	0.16	0.18
ED_{50}	-1	0.37	1.17	0.452
c	-0.81	0.44	1.02	0.02
$k1$	1.14	3.13	1.87	0.18
$k2$	0.23	1.26	1.2	0.22
x_0	5.26	192.48	0.22	0.717

ume where the dose was applied, denoted by $\bar{y}_p = 1057.42$ [mm^3] with standard deviation $\sigma_p = 614.45$ [mm^3].

During the generation, three conditions were imposed, that each θ_i-s must obey. The first condition is that for each θ_i the untreated tumor should grow, i.e., $a-n > 0$. The second condition filters those parameter sets for which the tumor is resistant and grows continuously for an initial 8 [mg/kg] dose. The last condition restricts the maximum deviation of the tumor volume between the volume at the injection and the peak tumor volume to $y_p \pm \sigma_p$. These rules ensure that the generated parameter sets lead to non-resistant tumors while they also share similar traits to their real-life counterparts. Using these restrictions, a virtual population $\mathcal{V}_r$ with 100 elements was generated that will be used for robustness analysis. A different population $\mathcal{V}_o$ was also generated with the additional condition, that each tumor should shrink under 10 [mm^3] between $t_r \pm \sigma_r$, from a random initial condition x_{10} with initial dose 8 [mg/kg]. The role of $\mathcal{V}_o$ is to provide virtual species for which a single dose can reduce the tumor completely so that one can easily compare the effectiveness of algorithmic therapies, attributed to different control parameters, with the MTD injection. The generated virtual populations can be seen in Figure 4.6.

In the case of the third iteration of the controller design, the virtual patients were generated in essentially the same manner. A minor difference was that instead of using the heuristic *scattersearch* algorithm of MATLAB, the FIE was employed to determine the initial conditions for the SAEM. Furthermore, obtaining more data from additional experiments does not require one to cut the time-series into different regions, as in the previous case. The obtained parameter values for 51 time-series can be seen in Table 4.4. Since the additional time-series contain significantly more doses due to the metronomic nature of these therapies, the pharmacokinetic values can be estimated more accurately, hence their covariances were not penalized in the beginning of the estimation.

4.2.9 Tuning the controller

To determine the control parameter r, a grid search was performed on a set of different values [C7]. For each value in the set, the controller was tested on each virtual patient in $\mathcal{V}_o$. Then, for all virtual patients, the doses applied in the simulations, and the final tracking errors at the last time instance were summarized. These two performance indices are then compared for the different controller values by forming a Pareto front.

The considered time span of the simulations was 100 days for each sample. During the tuning, the model parameters were assumed to be known precisely, and the full state measurement is available so that the use of MHE is omitted which alleviates the computational burden of the tuning procedure. The grid search was performed on the set of values $r \in 10^i, i \in \{-7, \dots, 3\}$ with $N = 4$ which is two weeks of prediction, with prediction intervals defined in (4.23). The setpoint was set to be $y_{\text{ref}} = 1$.

The grid search for the second iteration of the algorithm revealed that using smaller values for r lead to better convergence properties in each of the virtual patients, meaning less faulty iterations at each τ_i. For each r value the total amount of drug administered for the 100 patients was calculated in $\mathcal{V}_o$ and the sum of the tracking error at the end of the simulation interval, denoted by u_{sum} and e_{sum} respectively. One can see in Figure 4.7. that by decreasing the value of r, the total administered drug increases while the tracking error shrinks. It can be seen that the optimal choice, which results in the best trade off between the two metrics, is $r = 1$, for which $u_{\text{sum}} = 155$ [mg/kg], $e_{\text{sum}} = 176$ [mm^3] which is significantly better than the single 8 [mg/kg] dose case with $u_{\text{sum}} = 800$ [mg/kg], $e_{\text{sum}} = 414$ [mm^3].

4.2.10 In silico validation

The first controller was tested in silico, using model parameters obtained from [R37], same as in Chapter 3, Section 3.6. Each simulation was run for 350 days, resulting in 50 optimization instances on the whole interval, corresponding to weekly dosing [C5].

Using these simulation settings, the control parameters were empirically determined through simulations. First, the horizon length was determined by choosing a length long enough that the tumor is able to enter a remission phase. Using data from the first experiment, this was determined to be $N = 3$ with $\tau_{i+1} - \tau_i = 7$, resulting in a single dose weekly. For each model parameter combination, a different r value had to be tuned. The goal of the tuning was to find a particular r value that resulted in an initial dose smaller than the MTD. Fist, the value of $r = 10^0$ was chosen, after which the first dose was computed. If this does exceeded the MTD, then r was increased by an order of magnitude (e.g. 10^1), until the first computed dose was smaller than the MTD. Consequently, their values were chosen to be $r = \{10^5,\ 10^4,\ 10^6,\ 10^5,\ 10^4,\ 10^5,\ 10^5, 10^4\}$ for each subject respectively.

As one can see in Figure 4.9., the controller was able to handle the cases where the model parameters represented controllable systems. From the two uncontrollable cases, numerical errors were present in $S_{0,1}$, which required some modifications regarding the proposed algorithm, which will be shown in Section 4.2.6. Before the first experimental validation, the dynamical model was also replaced with the four-state dynamical model with the same control parameter configurations. Furthermore, the time between each consecutive injections were lowered to $\tau_{i+1} - \tau_i = 3$. Because PLD treatment is given through the tail vein of the animal, a minimum of 3 days recovery is required between drug injections. Technically, the frequency of tail vein administration should be limited to a minimum to avoid unnecessary stress ([R65]), and, additionally, the PLD treatment could cause inflammation and necrosis if administered more frequently. This also entails, that the length of the prediction horizon must be enlarged, which I chose to be $N = 7$ to compensate for the desired three-week prediction interval. The corresponding state-estimator for this controller was developed in [R51], where only a subset of the original model parameters was estimated, namely a, b, n and w. This algorithm was then used in the first experiment, which will be described in Chapter 5.

The second and third controllers were validated on the robust parameter set $\mathcal{V}_r$, containing 100 subject [C7]. Each simulation was performed for up to 350 days, under which the majority of the population was able to achieve reference tracking. However, in some cases, reference tracking was unsatisfactory due to the short prediction horizon. As a consequence, the parameter N was enlarged to $N = 8$, due to its role in the stability of the control, which solved some of the instances. The final solution was to change the control penalization factor to $r = 0.1$, which resulted in perfect tracking for each virtual patient.

The solution curves for the second controller can be seen in Figure 4.10.

For the third design a new robust parameter set $\mathcal{V}_r$ was generated using values from Table 4.4. The control parameter was also changed back to $r = 1$, since the doses applied during the in vivo experiment turned out to be too high which lead to toxicitiy in some subjects. The resulting in silico curves can be seen in Figure 4.11. Comparing the results with the previous case, two qualitative differences can be seen. In Figure 4.11, the maximum of the tumor volume is higher than in the previous case, since the control action is penalized more. Also, this effect can be attributed to the parameters of the virtual patients, since they were generated from additional data. Moreover, the steady state value of the tumor volume is more varied around the setpoint as in the former case. Using $r = 0.1$, the sum of absolute errors at $t = 350$ was 3.78 [mm^3], which for the $r = 1$ case was 1130 [mm^3]. Nonetheless, the experience from the in vivo validation of the second implementation indicates that choosing a larger control penalty is more beneficial due to the positive effect regarding systematic toxicity.

4.3 In silico validation of the NMPC-MHE scheme

The algorithms presented in this dissertation were tested in silico [C7] and in vivo [C6]. In the previous section, one could see in silico results for the controller on generated virtual populations, which showed their qualitative behavior. Before initiating an in vivo experiment to the end of validating the algorithm, the full proposed approach has to be tested in silico. This means, that the NMPC must be combined with the MHE, and additional effects must be included in the simulations, for example adding sensor noise and limiting the measurements and doses to specific days. In the current section, I will present the in silico validation of the full algorithm.

At the first implementation of the controller, a detailed sensor model and the virtual populations had not yet been available. As a consequence, the combined approach could not had been tested systematically through simulations before the first experiment, only their separate components, shown in Section 4.2.10 and [R51]. However, in the case of the second and third implementation of the controllers, the in silico validation was performed on the robust virtual population $\mathcal{V}_r$. For each parameter set in $\mathcal{V}_r$, the model was simulated with added sensor noise. Then, the MHE used these simulated measurements to determine the state estimates that the NMPC further utilized to compute the doses at the control time instances. This combined approach is visualized in Figure 4.12.

First, a noise model was developed to mimic the measurement errors arising from the caliper measurements. The noise model is taken from [C7], where a lowpass Butterworth filter was applied in a zero-phase setting. The noise is approximated as the difference between the raw, and the filtered time series. From these error terms, a histogram was created with five bins, each containing an approximately equal number of measurements. Then, the standard deviation of each bin is calculated, which can be accurately approximated by an affine function of the volume, which can be seen in Figure 4.13. The generated noise is assumed to be an additive Gaussian process in the form of

$$\begin{aligned} \sigma(y_k) &= 0.1 + 12.4 y_k, \\ \nu(y_k) &\sim \mathcal{N}(0,\, \sigma^2(y_k)), \\ \tilde{y}_k &= y_k + \nu(y_k), \end{aligned} \tag{4.30}$$

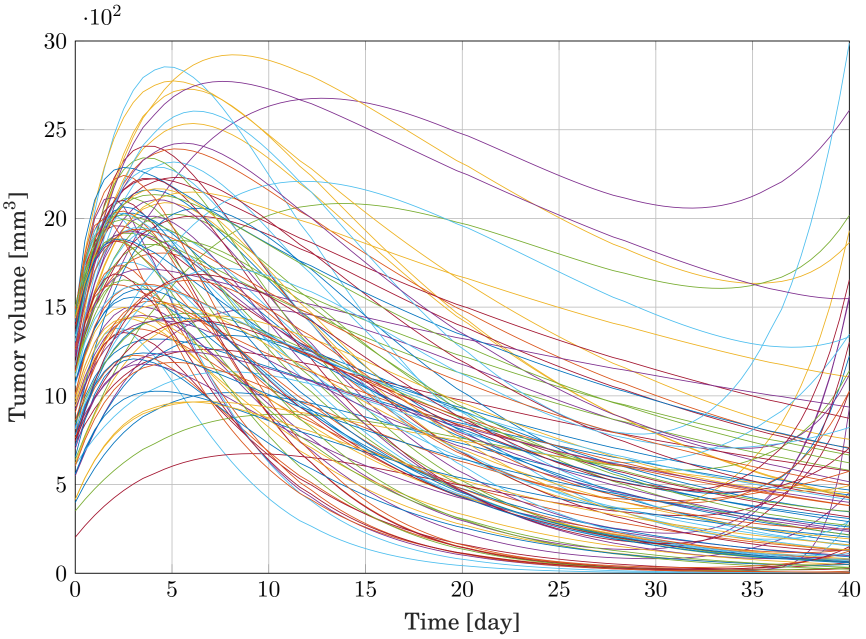
where y_k is the simulated output of the model at time t_k and $\tilde{y}_k$ is the output with added measurement noise. The exact values were computed from the time series of the baseline experiment, namely $S_{0,1} - S_{0,10}$.

For the second implementation of the controller, the robust virtual population was generated using the values in Table 4.3 on which its performance was evaluated. The controller was augmented with an MHE, implemented in [R51] to provide the state estimates. The control parameters were set to $r = 0.1$ and $N = 8$, as determined previously, with a reference $y_{\mathrm{ref}} = 1$ and maximum simulation interval 350. During a week, five measurements are assumed to be taken from Monday to Friday, and the administrations are applied twice per week, as described in (4.23). In total, the algorithm was tested on 25 virtual patients [C7].

The results of the simulation can be seen in the top part of Figure 4.14., where each tumor shrinks down to the neighborhood of the setpoint. On average, the virtual patients received 16.03 [mg/kg] dose during the simulation interval. Because measurement errors are introduced, it is not obvious to determine whether true setpoint tracking was achieved in this case. Nevertheless, at the end of the simulation, the largest tumor volume was 31.36 [mm^3], indicating a significant decrease in the tumor sizes compared to their initial value.

In the case of the third experiment, the same sensor model was used in the simulation study. Moreover, the MHE used in the previous experiment was replaced with the one developed in Chapter 4.1. The virtual population was also replaced using the updated parameter estimates, contained in Table 4.4. The parameters of the simulation and the controller remained the same, but the control penalization was modified to be $r = 1$ due to previous considerations on the generated doses in the second experiment.

The results of the simulation can be seen in the bottom part Figure 4.14. It can be immediately seen that the reduction of tumor volume was not as significant as in the previous case due to the larger penalty on the control action. Here, the maximal tumor volume at the end of the simulation was 231.47 [mm^3] and the virtual patients received 153.11 [mg/kg] drug during the treatment on average. Note that the amount of drug that is given during the therapy is significantly larger than in the previous case, while the final tracking error is also higher. This result contradicts the expected behavior of the increased penalization, namely the amount of drug should be smaller than in the previous case. This effect can be explained by comparing the values in the tables that were used during the generation of the virtual populations. In particular, the value of ED_{50} is notably higher in the second case, namely 0.3679, compared to 0.0012. Since ED_{50} is directly linked to x_3 and the applied dose d_i, higher parameter values yield an increase in the computed doses.



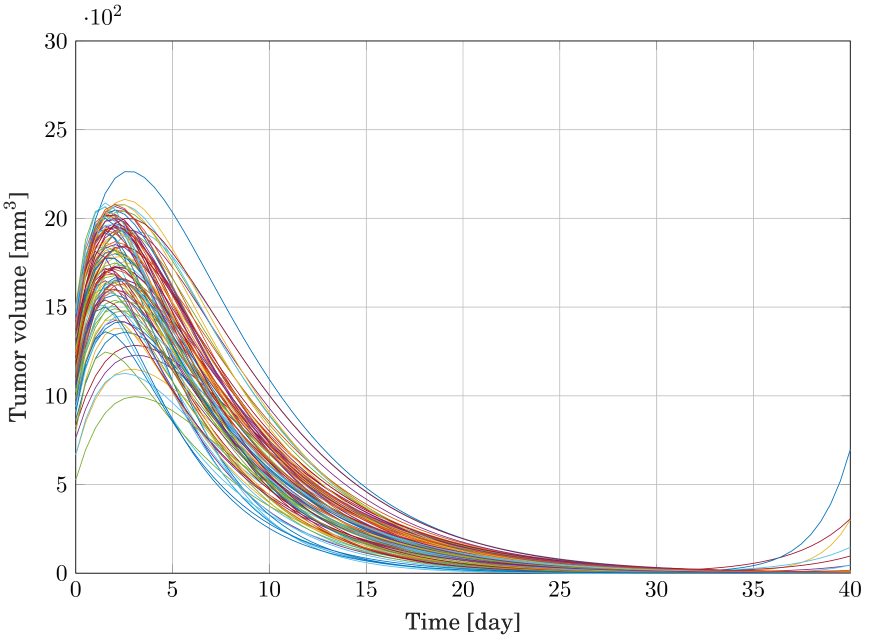


Figure 4.6: The two generated virtual populations $\mathcal{V}_r$ and $\mathcal{V}_o$ using the values from Table 4.3, subject to a single 8 [mg/kg] administration on the first day.

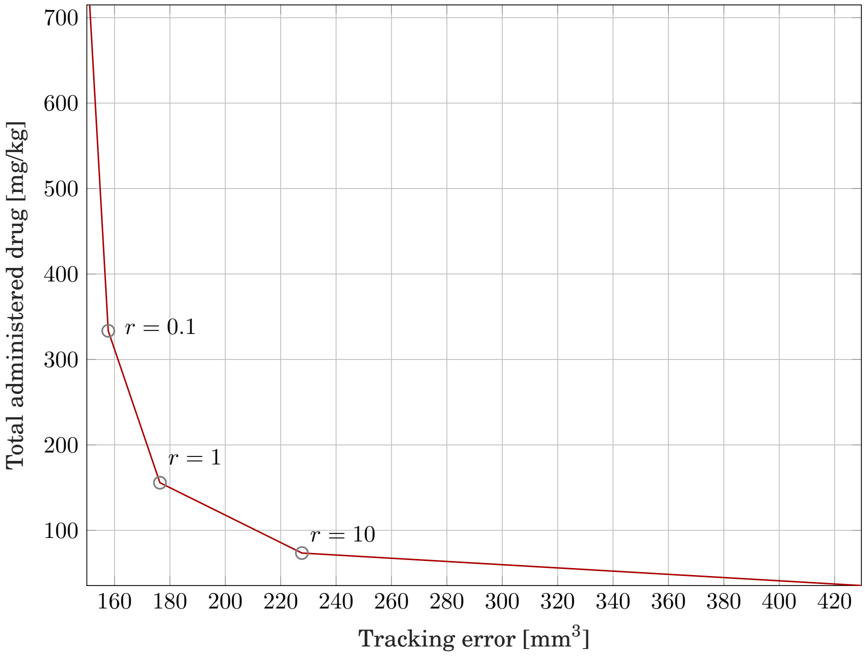
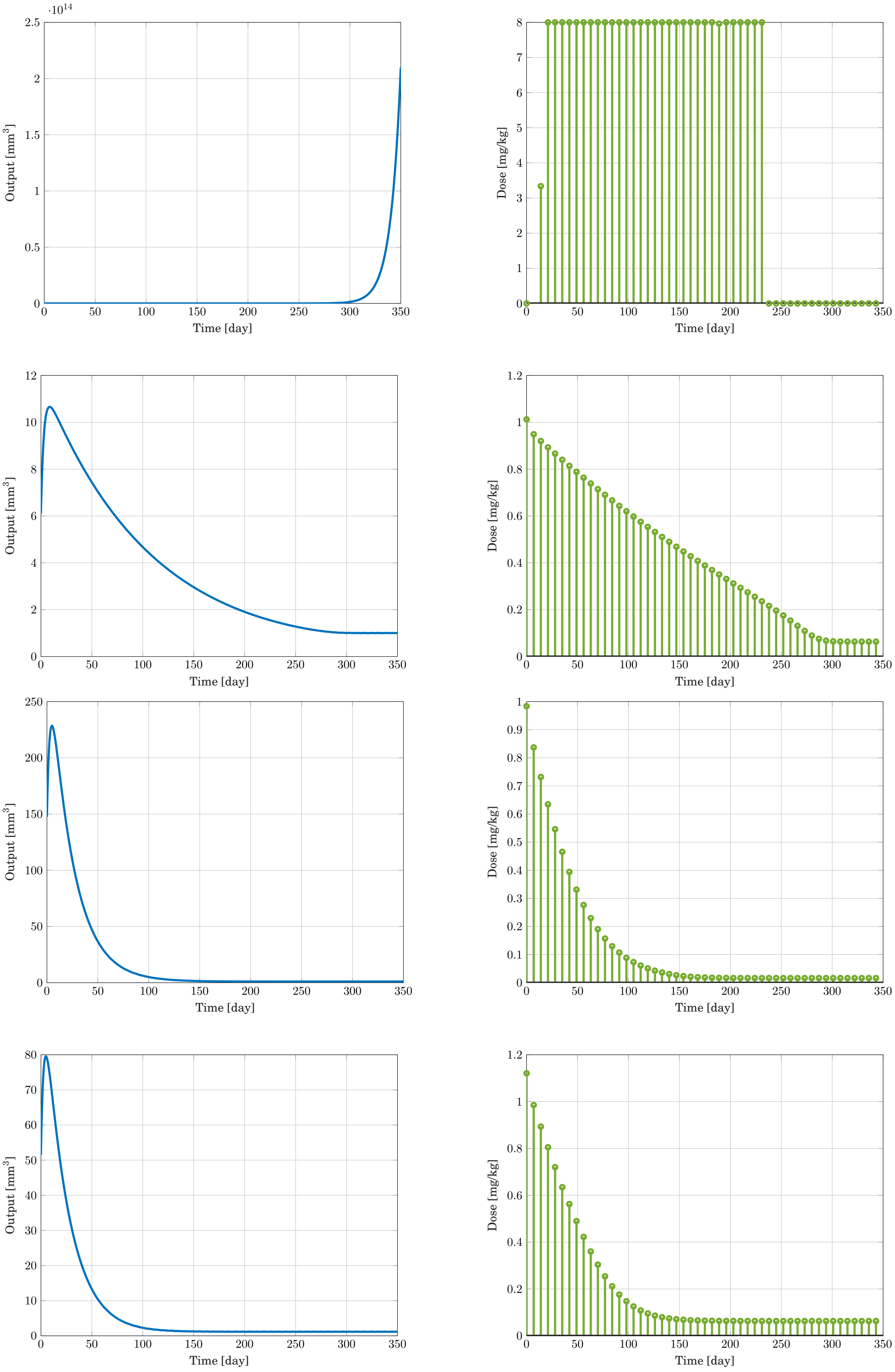


Figure 4.7: Pareto front of the grid search for varying r values for the optimal virtual population $\mathcal{V}_o$, generated from the values listed in Table 4.3.


·10^14
2.5
2
1.5
1
0.5
0
Output [mm^3]
Time [day]
0
50
100
150
200
250
300
350
8
7
6
5
4
3
2
1
0
Dose [mg/kg]
12
10
8
6
4
2
1.2
1
0.8
0.6
0.4
0.2
250
200
150
100
50
0.9
0.7
0.5
0.3
0.1
80
70
60
40
30
20
10

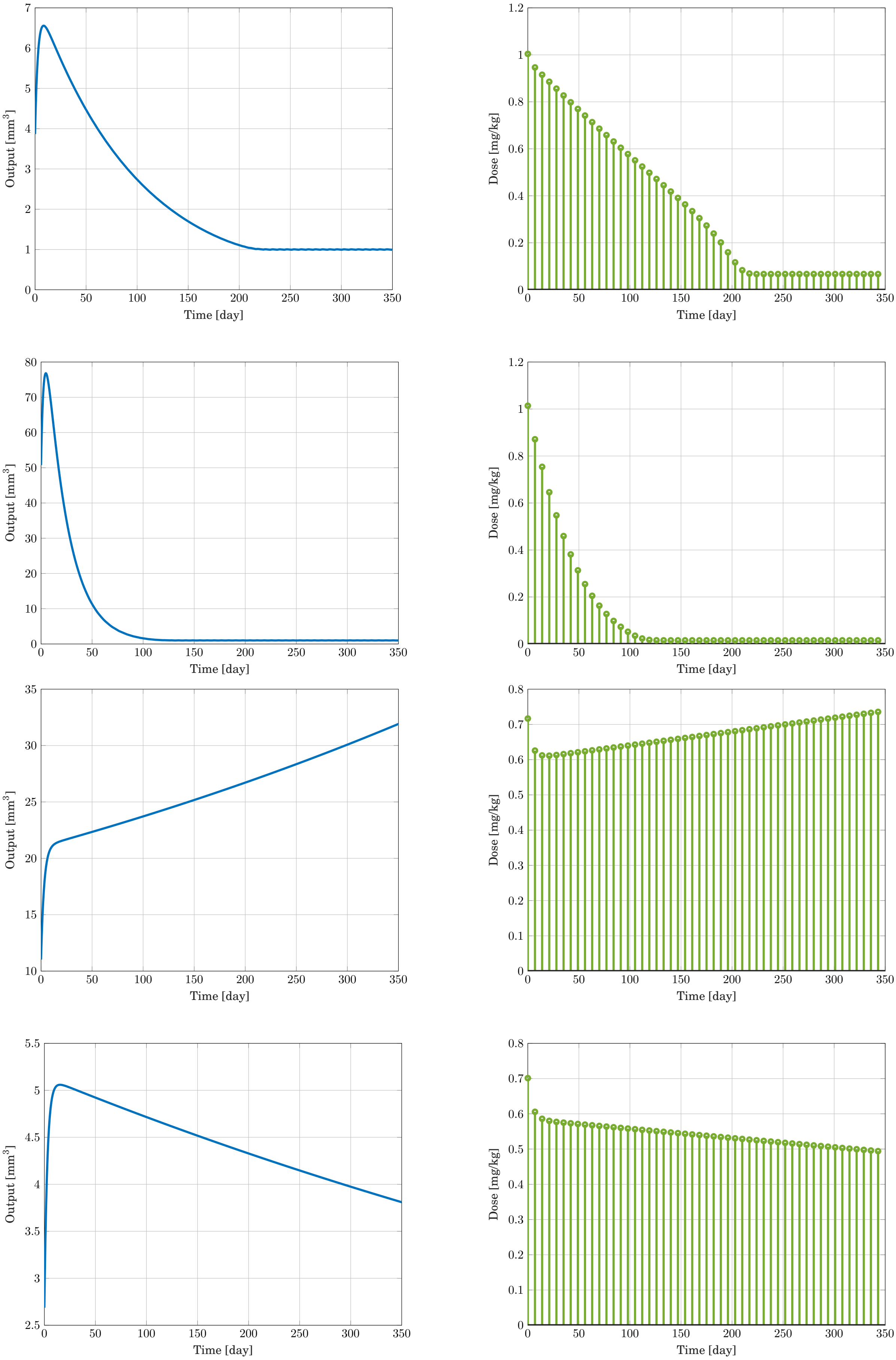


Figure 4.9: Results of the optimization for $S_{0,1},\ S_{0,2},\ S_{0,3},\ S_{0,4},\ S_{0,5},\ S_{0,6},\ S_{0,8}$ in a descending order from top to bottom.

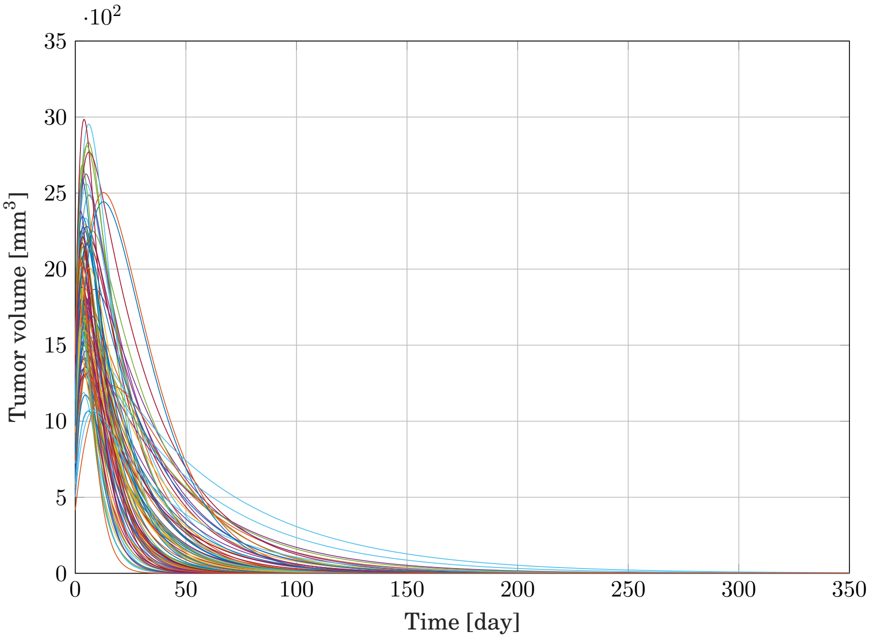


Figure 4.10: Resulting tumor volumes for the robust dataset $\mathcal{V}_r$ for the second implementation of the controller.

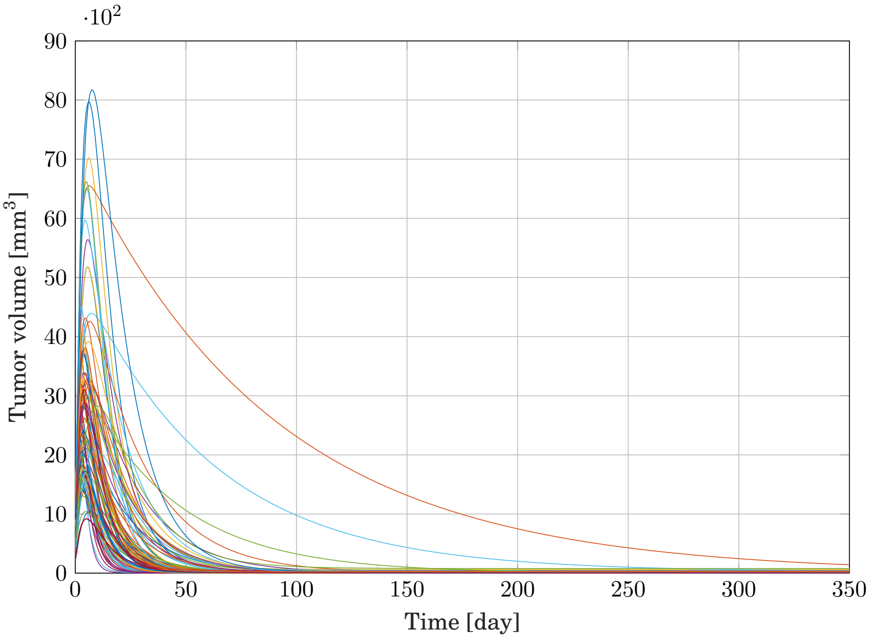


Figure 4.11: Resulting tumor volumes for the robust dataset $\mathcal{V}_r$ for the third implementation of the controller.

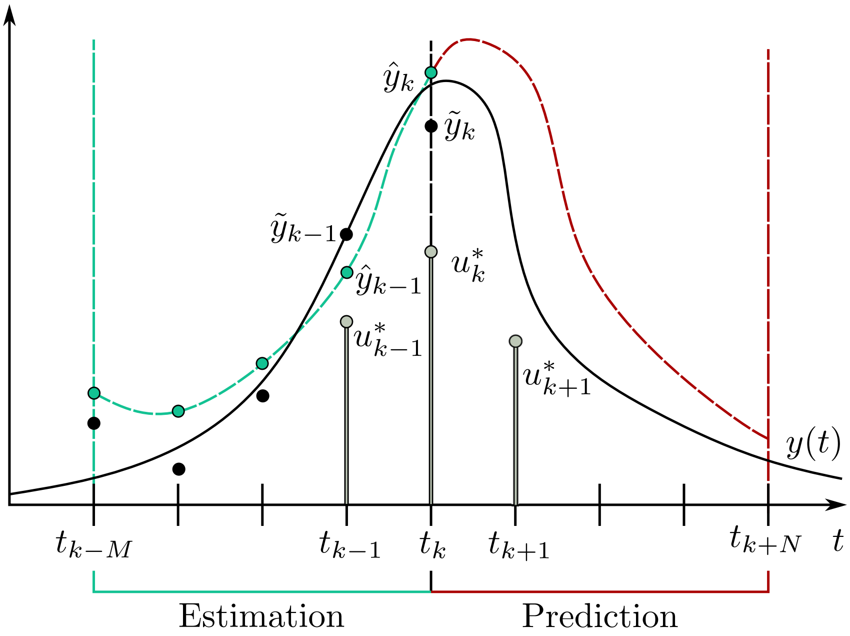


Figure 4.12: Visualization of the full control loop. The black curve is the evolution of the tumor volume, the turquoise curve is the estimated tumor volume provided by the MHE, and the red curve is the prediction of the NMPC, based on the optimal sequence of administrations $\boldsymbol{u}^*$. Black dots ($\tilde{y}_k$) denote the volume measurements, turquoise dots ($\hat{y}_k$) denote the estimated volumes, and t_k denotes measurement time instances. In this diagram, $t_k = \tau_i$, such that measurements and administrations are performed on the same day.

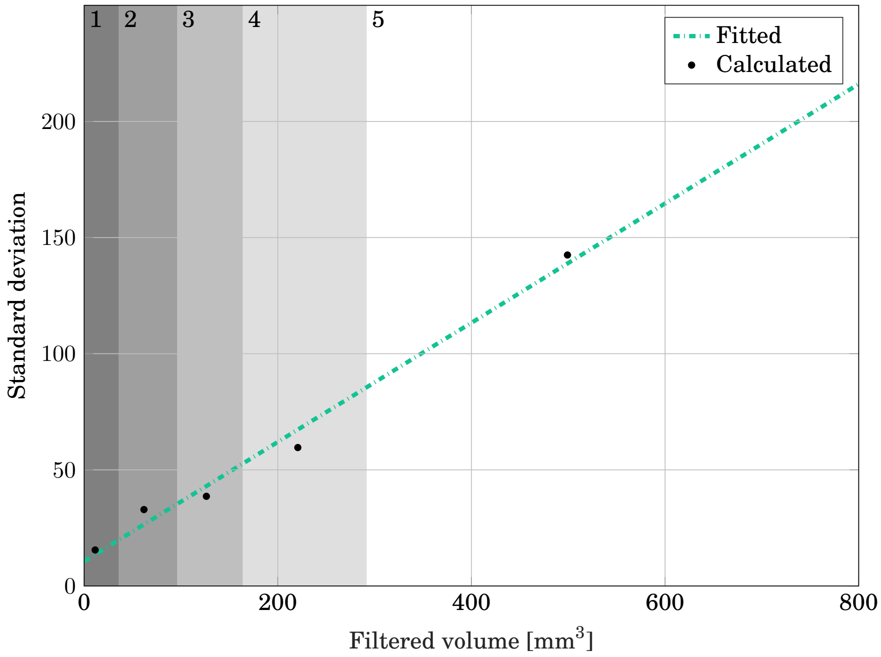
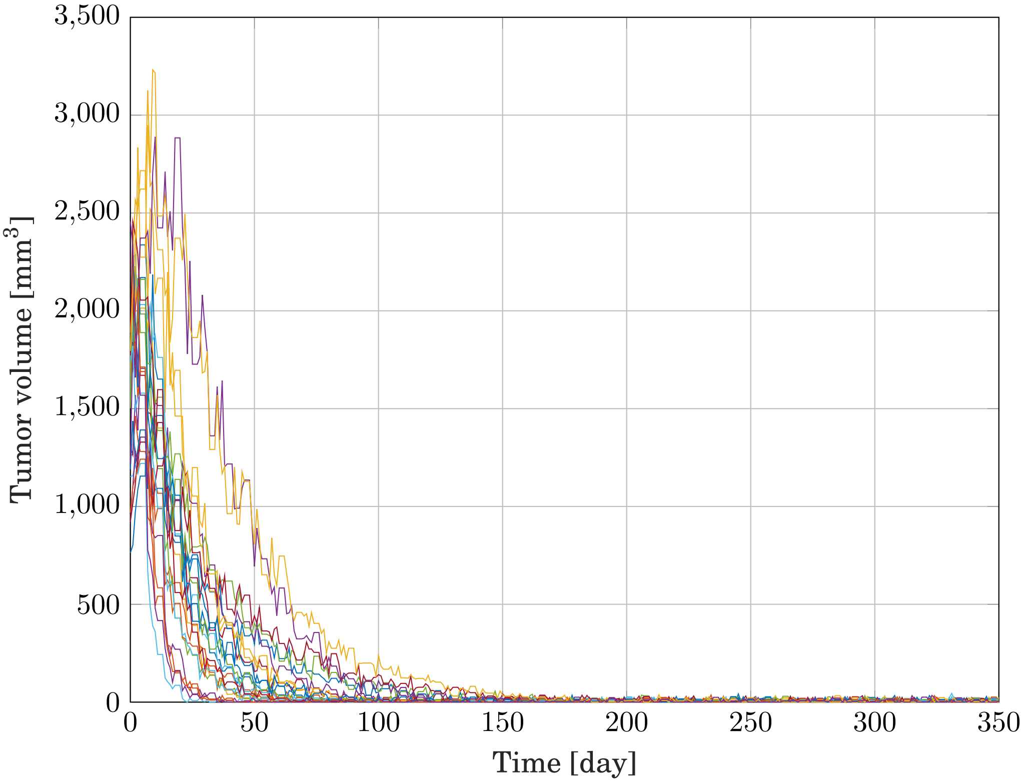


Figure 4.13: The calculated standard deviations and their approximation. Each shaded region from 1 to 5 represents a bin, where the deviation was calculated.



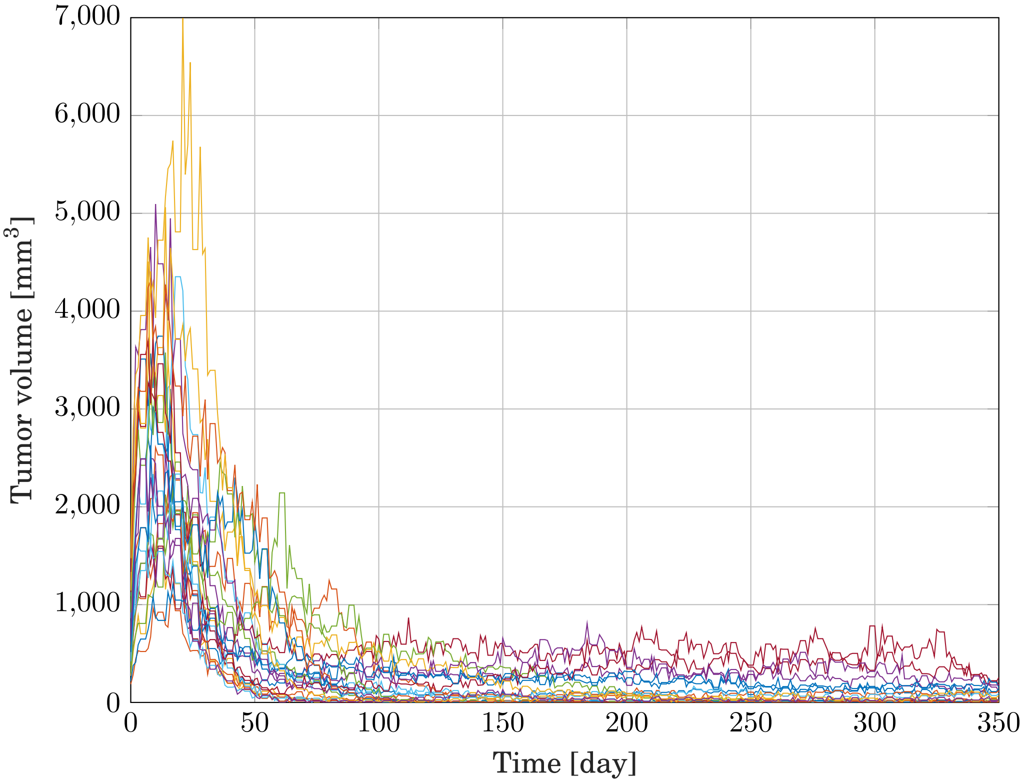


Figure 4.14: In silico results for the second and third implementation of the controller on the robust set generated from Table 4.3. and 4.4.

5

Experimental validation

Thesis Group 3: In vivo validation of the control algorithms

Thesis 3
I tailored the algorithms for in vivo experimental validation using mice. The first two iterations of the algorithm were tested in separate mice experiments, which indicated that the proposed schemes can also be viable in practice.

Publications relevant to the theses: [C6, C7, C5]

Thesis 3.1
I generated closed-loop therapy protocols of the first iteration of the NMPC algorithm for in vivo validation. Results showed that the closed-loop approach is able to achieve remission in the subjects without using excessive amount of drug during the therapy.

The results can be seen in Section 5.2.

Thesis 3.2
I implemented the second iteration of the NMPC controller for closed-loop in vivo validation. During the experiment, I have generated the optimal dosages for the mice using the algorithm. Results showed that mice treated with the algorithm had similar mean survival as the conventionally treated subjects, however, some mice reached significantly longer survival.

The results can be seen in Section 5.5.

In the current chapter, the in vivo validations of the designed controllers are presented. Firstly, the baseline dataset is introduced, where the standard protocol of chemotherapeutic administration was used to treat 10 subjects with PLD. Consequently, the experimental setup, objective, and results of the controller's first, second, and third implementations are shown. The baseline experiment is referred to as $\boldsymbol{S}_0$, while the first, second and third are with $\boldsymbol{S}_1, \boldsymbol{S}_2$ and $\boldsymbol{S}_3$ consecutively. A given mouse in an experiment is denoted by $S_{i,j}$, where i is the experiment, while j is the subject identifier. Currently, there are three finished experiments (including the baseline), and one is ongoing.

Concerning the in vivo validation, all animal housing and breeding processes and experimental protocols were approved by the Hungarian Animal Health and Animal Welfare

Directorate according to the EU's most recent directives. All surgical and treatment procedures were performed according to the Committee on the Care and Use of Laboratory Animals of the Council on Animal Care at the Institute of Enzymology, Eötvös Loránd Research Network in Budapest, Hungary (001/25746/2015).

5.1 Baseline dataset

In the baseline experiment, denoted by S_0 and described in [R25], no algorithmic protocol was employed. This experiment aimed to compare the multi-drug resistance properties of PLD to regular doxorubicin using the standard protocol. In the standard (or conventional) protocol, mice are dosed when the tumor grows above $200[\mathrm{mm}^3]$ in a subject (called the tumor trigger). When the volume is above the trigger, the mice receive the MTD (8 [mg/kg] dose) every 10 days until the tumor volume is reduced to 50% of its original value. In the case of the tumor trigger, the volume is calculated as

$$y_i = y(t_i) = \mathrm{length}(t_i) \cdot \frac{\mathrm{width}(t_i)^2}{2}, \tag{5.1}$$

which is a standard method to calculate the volume in mice experiments, yet less accurate than the formula shown previously in (2.1) according to [R35]. In order to monitor the state of the tumor, measurements were taken at least three times per week, once they became tangible. During the experiment, 10 mice were administered with PLD.

While in this experiment, no algorithm was validated, many of the results presented in the dissertation used these data, hence I found it essential to include it in the thesis as a separate entity. Firstly, this data set was used initially to identify the parameters of the three states model in [R37]. These identified parameters were then employed to validate the RFPT method in Chapter 3. Secondly, this dataset was used to generate the first virtual population to tune the NMPC for the first experiment S_1. Additionally, the standard therapy used in this study can be used to compare the algorithmic therapies validated in the following experiments. The full dataset can be found in [R25].

5.2 First experiment

The goal of the first in vivo validation of the controller was to validate whether the algorithm is capable of decreasing the tumor volume using less amount of drug than the standard therapy [C6]. The controller was described in Section 4.2.5., while the MHE used in the study was implemented in [R51]. Chronologically, the model with four states was developed a few weeks before the experiment was initiated. Since this model could be identified more accurately, the three-states model was replaced in the NMPC for this variant. This only caused a change in the artificial initial conditions, which became $\boldsymbol{s}_{i+l} = [y_0,\ 0,\ 0,\ 0]$.

In the first experiment S_1, 41 mice were divided into four distinct groups. When the tumor trigger became active for the first time, the mice received a dose of $0.5,\ 1,\ 4,$ or 6 [mg/kg], according to the group to which they were assigned. Five minutes after this first dose, three mice were sacrificed from each group (in total 12) to measure the serum level of the drug using mass spectrometer. The rest of the mice were then assigned randomly to four different groups, in which they received therapies based on different algorithmic considerations. Since these therapies are based on different algorithms, they were excluded from the following discussion. Moreover, due to the limited number of mice, there was no control group in this study.

Six mice were assigned to the group regarding the closed-loop NMPC-MHE therapy, denoted by $S_{1,j}$ with $j = 1, 2, \ldots, 6$. From these subjects, $S_{1,1}$, $S_{1,2}$, $S_{1,3}$ received a 4 [mg/kg] initial dose and $S_{1,4}$, $S_{1,5}$, $S_{1,6}$ got 6 [mg/kg].

For these mice, the entire course of the experiment is described here. First, the experiment started with the implantation of cancerous cells in mice. When the tumor trigger became active, the mice received a single dose of either 4 or 6 [mg/kg] for identification. A day before the algorithm's first administration was computed, the model parameters of the tumor with four states were identified (after estimating the pharmacokinetic parameters) using the SAEM algorithm solely, which can be seen in [C6]. The initial plan was to initiate the algorithmic therapy individually when the tumor trigger becomes active again. However, after 23 days, the therapy started for each mouse regardless of the size of their tumor, due to a request by the staff in the animal house such that the logistics of the experiment could be minimized. The therapy then lasted for 27 days, under which 9 doses were computed. The administrations took place every Monday and Thursday (except the first dose computed on Friday) for four weeks. After 32 days, the algorithmic therapy was stopped, and the experiment concluded when each mouse passed away. The experiment occurred between August 8, 2020, and January 15, 2021, and the doses were generated between September 4, 2020, and October 1, 2020. During the 27 days of the automated therapy, subject $S_{1,4}$ was in complete remission, hence it is omitted from the following discussion.

For each mouse, the first task was to tune the controller, since at this point, the virtual population-based tuning approach had not been implemented. Consequently, the control penalization r had to be tuned individually for each identified parameter set in the same manner as in Section 4.2.10. The r value for mice $S_{1,1}$, $S_{1,2}$, $S_{1,3}$, $S_{1,5}$, and $S_{1,6}$ was found to be $10^7, 10^5, 10^7, 10^6$, and 10^6 respectively. The reference volume to be tracked was $y_{\text{ref}} = 1$, and y_0 was the tumor volume measurement on the first day of the closed-loop trial.

During the experiment, the value of r was increased by an order of magnitude in the case of $S_{1,1}$, $S_{1,5}$, and $S_{1,6}$. This change was performed to avoid cumulative toxicity, which was not accounted for explicitly in this version of the controller. In particular, for $S_{1,1}$, the control parameter was set to $r = 10^8$ on September 8. For $S_{1,5}$, r was changed to 10^7 on September 10 and 10^8 on September 24. Finally, the r value of $S_{1,6}$ was modified to $r = 10^7$ on September 10.

Results of the experiment for subjects $S_{1,1}$ and $S_{1,1}$ can be seen in Figures 5.1 and 5.2, while the other time series are presented in Figures B.1, B.2 and B.3 in Appendix B. In each figure, the first plot shows the interpolated evolution of the tumor (blue line), calculated using (2.1), with the actual measurements (red crosses) and their estimates, produced by the MHE (black dots). The second plot shows the corresponding dosing schemes, where the first dose is used to identify the parameters, and the controllers generate the remaining doses. The last plot shows the time-varying parameters computed by the MHE.

It can be clearly seen that the generated protocols can induce remission in the subjects, invariant to the tumor size on the first day of the treatment, similar to the standard protocol. In each case, the tumor volume shrank under $200[\text{mm}^3]$ during the 27 days of treatment. This means that the first objective of the validation was successful, even though human intervention had to be utilized during the experiment.

Regarding the initial goal of the experiment, it is not a simple task to determine whether the algorithm uses less drug than the standard protocol. A possible way is to compare the sum of doses in the one month interval produced by the standard and algorithmic protocol. In the baseline experiment (standard protocol), each initial dose reduced the tumor volume ultimately. This means that injecting the MTD when the tumor trigger is initially activated leads to complete remission in every case, and the tumor shrinks below the trigger. However, when the tumor trigger became active again, additional doses had a weaker effect and had to be repeated at least two times (except for $S_{0,9}$, referred to as PLD9 in the original article) to achieve the same effect. Consequently, under the 27 days of treatment, this would yield a maximum of 16 [mg/kg] (twice the MTD) PLD in the case of the standard therapy.

Table 5.2. summarizes the total doses given through the experiment, and the time when the subject was terminated, denoted by d_{sum} and $t_{survival}$, respectively. By subtracting the correct initial dose (4 or 6 [mg/kg]) from d_{sum} in the case of each mouse, the result is smaller than 16 [mg/kg]. Nevertheless, this comparison could be improved by accounting for the substantial level of inter-patient variability, which affects the effectiveness of the doses.

Concerning the parameter estimation, only the model parameters $a,\ b,\ n$, and w were estimated, as mentioned in Section 4.1.1. It can be seen from the diagrams that while $a,\ b$, and n show a significant variation in their values, the washout term w stays essentially constant. The variation might be attributed to the s.l.i. property of the parameters, as explained in Subsection 4.1.1. Nevertheless, the estimates could follow the measurements with only a negligible error, as seen in the parameter estimation figures.

The first experimental validation showed that the algorithm had several flaws that had to be corrected. The first and most severe issue was that the algorithm had to be tuned manually to decrease the amount of drug given to the subjects. In order to overcome this issue, a constraint was built in the next iteration that accounts for the sum of the doses given in the past days, as described in Section 4.2.6. Additionally, this can also be associated with the empirical tuning of the controller, for which the tuning process described in Chapter 4.2. was developed. The experiments also had some limitations, including the length of the automated therapies (which was only a month long) and the lack of a control group. Moreover, the comparison between the standard therapy and the algorithmic was not straightforward due to the short time interval in which the administrations took place.

5.3 Second experiment

In the second experiment, the goal was essentially the same as in the case of the first experiment, i.e., validating whether the controller can induce significant remission using less drug than the standard therapy. In this case, the second version of the controller was tested, which was previously introduced in Section 4.2.6. At this point, the virtual population-based tuning and validation were established (shown in Section 4.2.10.), which was run before the in vivo experiment. The corresponding MHE was the same as in the previous experiment, only with minor alterations. Additionally, the experimental protocol was modified so that the obtained results could be more easily compared with the standard protocol.

In $\boldsymbol{S}_2$, there were 49 mice in total. Each mouse was assigned into one group $\boldsymbol{G}_1$ or $\boldsymbol{G}_2$, where they received an initial dose of either 4 [mg/kg] (in $\boldsymbol{G}_1$) or 6 [mg/kg] (in $\boldsymbol{G}_2$) for parameter identification when the tumor trigger became active. In $\boldsymbol{G}_1$ and $\boldsymbol{G}_2$ there were 22 and 27 mice respectively. Here, no smaller initial dose was injected (i.e., 0.5 or 1 [mg/kg]) since they were found to be ineffective in the first experiment. Additionally, no mice were sacrificed after the initial dose, unlike in the first experiment.

After the initial dose, each mouse was randomly assigned to one of three groups. The first group received the NMPC-MHE therapy and had 21 mice in total, denoted by $S_{2,j}, j = 1, 2, \dots, 21$. The second group also had 21 mice, which received a therapy based on different algorithmic considerations, and thus it was omitted from the following discussion. The third group was a control group $\boldsymbol{S}_{2c}$, and had 7 mice, denoted by $S_{2c,j}, j = 1, 2, \dots, 7$. The control group received the standard therapy, with the exception that instead of using the MTD (8 [mg/kg]) for each administration, only 6 [mg/kg] was injected into the mice. This is because, in the algorithmic therapies, the same 6 [mg/kg] hard constraint was used due to frequent administrations.

In the case of the NMPC-MHE group $S_{2,j}$, the mice $S_{2,1}$-$S_{2,11}$ received 4 [mg/kg] initial dose, while the remaining mice $S_{2,12}$-$S_{2,21}$ were subject to an initial dose of 6 [mg/kg].

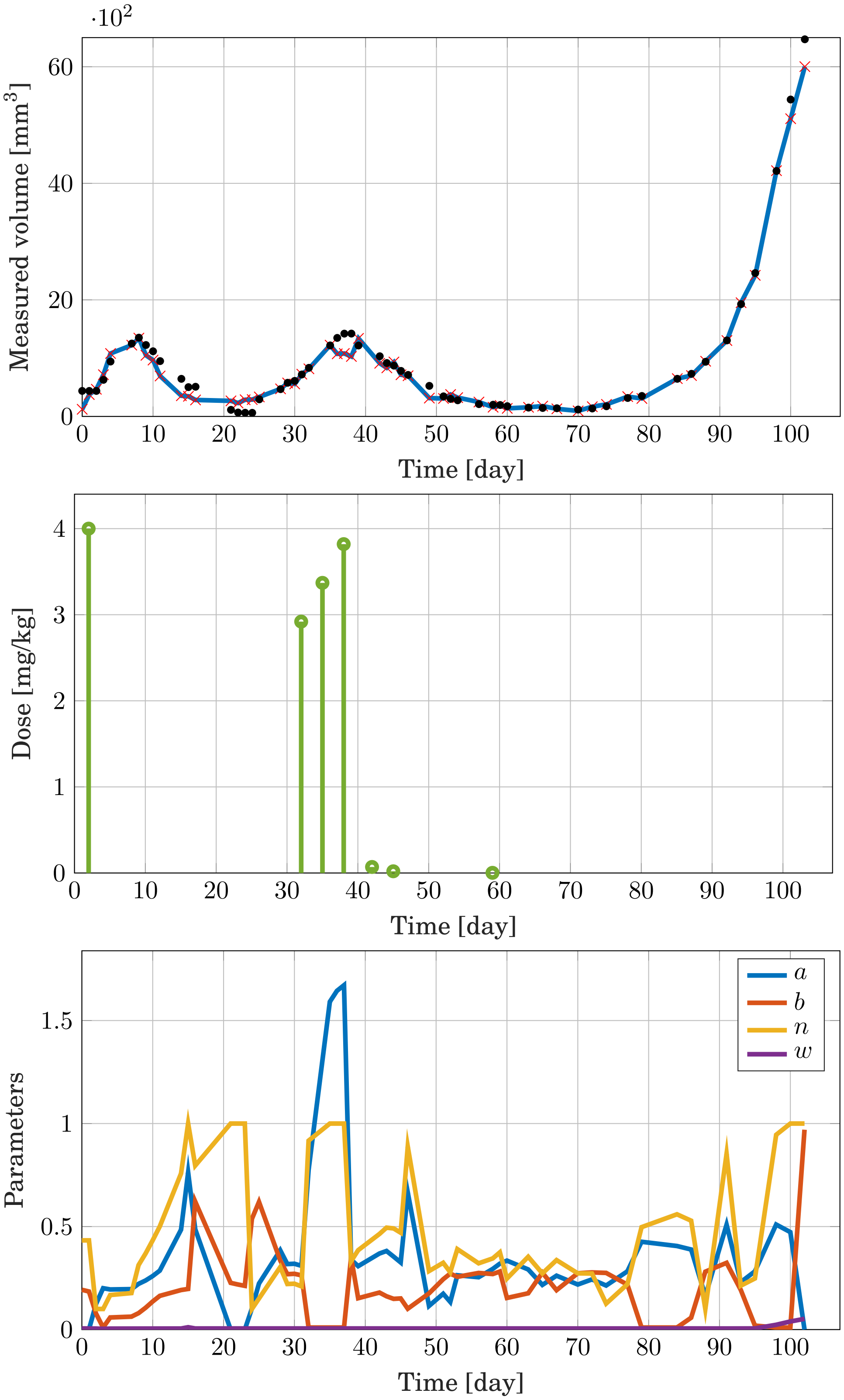


Figure 5.1: Evolution of the tumor volume, the corresponding doses, and the identified time-varying parameters for $S_{1,1}$. The red crosses represent the tumor volume measurements, while the black dots show their estimations produced by the MHE.

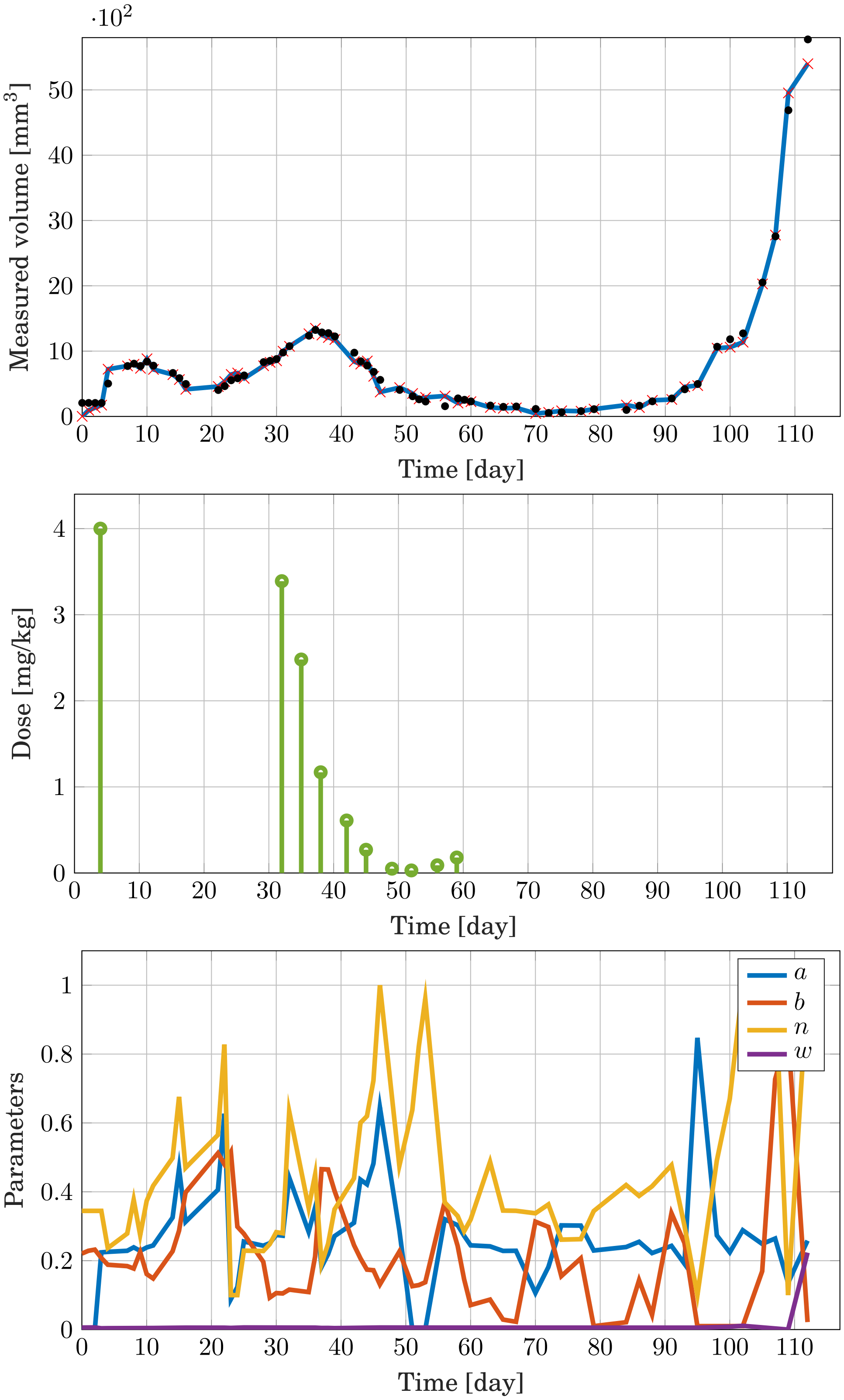


Figure 5.2: Evolution of the tumor volume, the corresponding doses, and the identified time-varying parameters for $S_{1,3}$. The red crosses represent the tumor volume measurements, while the black dots show their estimations produced by the MHE.

After the initial dose, the algorithmic therapy was initiated when the tumor trigger became active again. Before the first administration, the SAEM algorithm was run to obtain individual parameter fits. In this case, the MHE only used the population average as the initial parameter guess for each mouse, which was $\boldsymbol{p} = [0.71,\ 0.94,\ 0.13,\ 0.23,\ 0.0016,\ 0.52,\ 65.73,\ 71.21]$.

Each mouse was dosed twice a week on each Tuesday and Friday. The experimental data spans March 24, 2021, and March 7, 2022. The algorithmic therapy started on April 13, 2021, and lasted until December 10, 2021, or the termination of the given subject. The reason for halting the experiment on December 10, 2021, was that while five subject was still alive, their physical condition was poor due to the emergence of systematic toxicity. The results of the experiments can be seen in Figure 5.3., 5.4 and 5.5, while the remaining can be found in Appendix B., between Figures B.4. and B.21.

The results show that the applied doses are much higher than in the previous experiment. Since one experiment goal was to test the fully automated algorithm, no adjustments were made to the control parameters during the experiment. The leading cause of death was cumulative toxicity for most mice, even with the hard constraint $\boldsymbol{d}^c$ in (4.21). In particular, for subjects, $S_{2,2}$, $S_{2,4}$, $S_{2,6}$, $S_{2,8}$, $S_{2,9}$, $S_{2,12}$, $S_{2,14}$, $S_{2,16}$, $S_{2,19}$, $S_{2,20}$, and $S_{2,21}$ the cumulative toxicity leads to a sudden decrease in their mass, hence they were sacrificed to avoid suffering of the animal. The most severe side effect was observed in the case of $S_{2,5}$, where between days 76 and 132, the therapy had to be stopped until the subject recovered from its adverse condition. As previously mentioned, the algorithmic therapy was stopped on day 261, since in the remaining living mice $S_{2,5}$, $S_{2,10}$, $S_{2,11}$, and $S_{2,15}$ the tumor volume was already minimal and their physical condition was poor. For subjects $S_{2,1}$, $S_{2,3}$, $S_{2,7}$, $S_{2,13}$ and $S_{2,17}$ one can observe drug resistance, as additional doses did not have any effect on the evolution of the tumor volume. Table 5.4 shows the total doses and survival times of each subject.

It can also be seen from the diagrams that in complete remission, there is a considerable variation between the doses given to a subject, whereas the tumor volume stays relatively constant. This can be attributed to two factors: the control parameter of the NMPC and the MHE strategy implemented in this algorithm.

Since in the second implementation, the control penalty was chosen to be $r = 0.1$ based on the in silico results (as described in Section 4.2.10.), even small measurement errors could lead to significant changes in the computed doses. Moreover, during the experiments, the slightly modified MHE had a high variation in the estimated model parameters between each measurement. Since these parameters are used in the NMPC optimization and affect the calculated doses, they strengthen the effects of low penalization. In each case, there is a significant variation between each consecutive estimated parameter, and in many cases, their values jump between the endpoints of the box constraint. Additionally, these effects then propagate to the estimated tumor volumes, which are not able to adequately track the measurements qualitatively.

For example, in Figure 5.4, there are two peaks in the estimated volumes (top figure, black dots), which shows a significant time delay in the estimations. This time delay was also present in subjects $S_{2,3}$ and $S_{2,18}$. Also in Figure 5.5., while there is no time delay, the estimator was not able to adequately estimate parameters when drug resistance starts to emerge after day 120. The same effect can be observed in the case of $S_{2,1}$, $S_{2,13}$, and $S_{2,17}$. For the remaining subjects, where significant remission was achieved, the MHE was able to capture the time evolution of the measurements, but the estimated parameters showed high variations in these cases as well.

Convergence issues in the NMPC were also present during the experiment, which was also associated with the scaling term of the tumor volume in the cost function. This was chosen to be the tumor volume at the beginning of the algorithmic therapy since it was assumed that no apriori information was available on the maximum value of the tumor volume. Nonetheless, this assumption was refined later since if the tumor exceeds a particular volume, the mice must be sacrificed, otherwise, it is considered animal cruelty. This means that the tumor volume will always move in a fixed domain, which can be used to normalize the data, leading to better optimization convergence. There should also be a maximum limit on the tumor volume in humans since the tumor can not grow indefinitely without killing the patient.

5.4 Third experiment

The aim of the third experiment was also to test whether the algorithm could induce remission in the tumors through the computed doses using less drug than the standard protocol. Additionally, a secondary goal was to investigate if the additional changes in the algorithm led to a positive outcome regarding its performance. The controller in this study was based on (4.29) in Subsection 4.2.7., and the corresponding MHE was implemented in Section 4.1.4. The experimental protocol was also slightly modified in this case, such that it excludes the identification dose and permits the possibility of only a single dose per week.

There were 57 subjects in the third experiment $\boldsymbol{S}_3$. Each animal was assigned to one of three groups. In the first group, there were 21 mice in total, denoted by $S_{3,j}, j = 1, 2, \ldots, 21$, and they all received the NMPC-MHE therapy. In the second group, there were also 21 mice, and they received therapies based on a different strategy, thus they were excluded from the following discussion. The third group was the control group $\boldsymbol{S}_{3c}$ which had 12 mice in total, denoted by $S_{3c,j}, j = 1, 2, \ldots, 12$, and they received the standard therapy without any modifications.

An essential change in the experimental setup was that, in this case, no identification dose was given prior the algorithmic treatment. This is a critical step in adapting the algorithm to human use, since, in a realistic scenario, one should immediately start therapy when a tumor is detected in the patient. The model parameters used in the MHE, were then estimated from the previous experiments ($\boldsymbol{S}_0, \boldsymbol{S}_1$, and $\boldsymbol{S}_2$) based on the methodology described in Section 4.1.4, and their values can be found in Table 4.1. As indicated in the in silico tests, the control penalization term was also reset to $r = 1$.

The algorithmic therapy was initiated individually when the tumor trigger became active for the first time. Each mouse received administration on each Tuesday and Friday. After day 65, the frequency of dosing was altered based on the tumor volume. When the volume was under the tumor trigger for two consecutive weeks in a given subject, only a single dose was administered weekly on Tuesday afterwards. If the tumor trigger was activated again (due to the insufficiency of a single weekly dose), the administration continued twice per week. The reason for this change was that the frequent administrations ruined the tail veins of the mice in the previous experiment.

The experiment was initiated on June 10, 2022, and is currently ongoing. At the time of writing the thesis, $S_{3,17}$ is still alive from the NMPC-MHE group. In Figures 5.6., 5.7., and 5.8. one can see the status of the experiment up to day 386, while the time series associated with the remaining subjects are presented in Appendix B. between Figures B.22., and B.39. Here, the labeling is omitted from the parameter estimation diagram since it restricts the visibility of the results. For the parameters, a, b, n, w, ED_{50}, and c, the colors blue, red, yellow, purple, green, and turquoise are assigned, respectively. The parameter estimation diagram does not contain the curves for k_1 and k_2, which were also estimated. The underlying reason is that they remained constant in each case and had larger values than the other parameters, so the variation of the rest of the parameters could have been hard to comprehend in the diagrams.

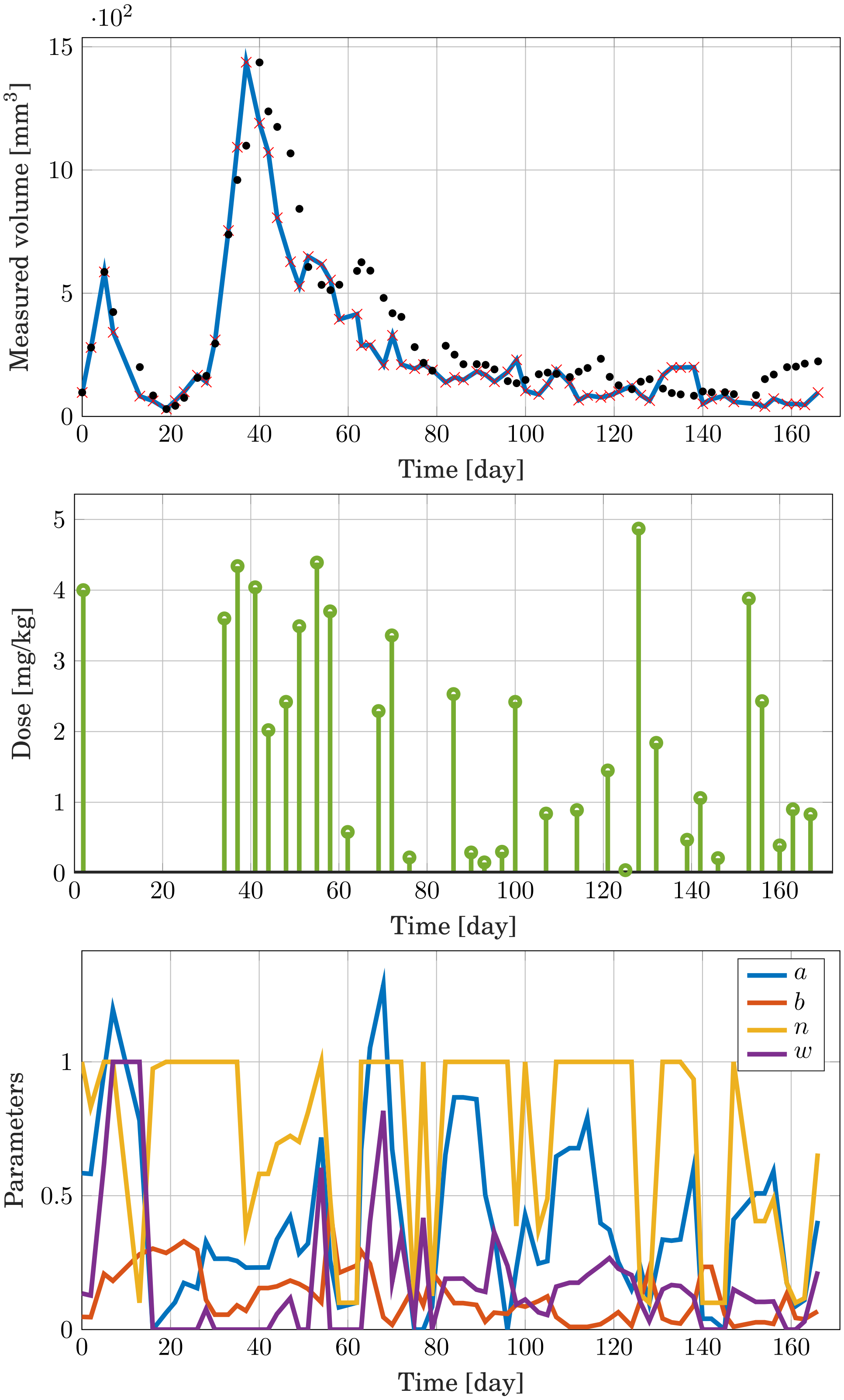


Figure 5.3: Evolution of the tumor volume, the corresponding doses, and the identified time-varying parameters for $S_{2,2}$. The red crosses represent the tumor volume measurements, while the black dots show their estimations, produced by the MHE.

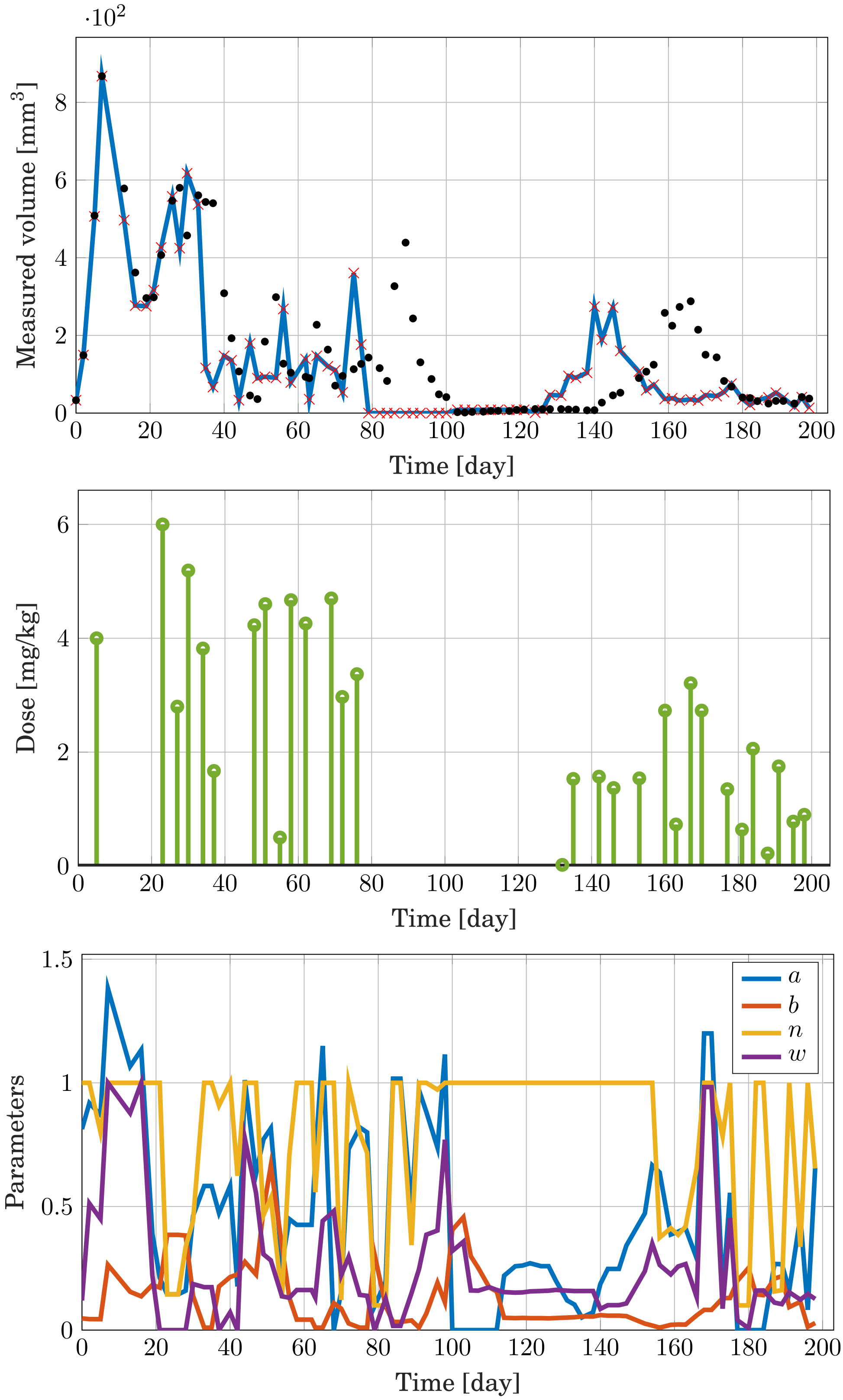


Figure 5.4: Evolution of the tumor volume, the corresponding doses, and the identified time-varying parameters for $S_{2,5}$. The red crosses represent the tumor volume measurements, while the black dots show their estimations, produced by the MHE.

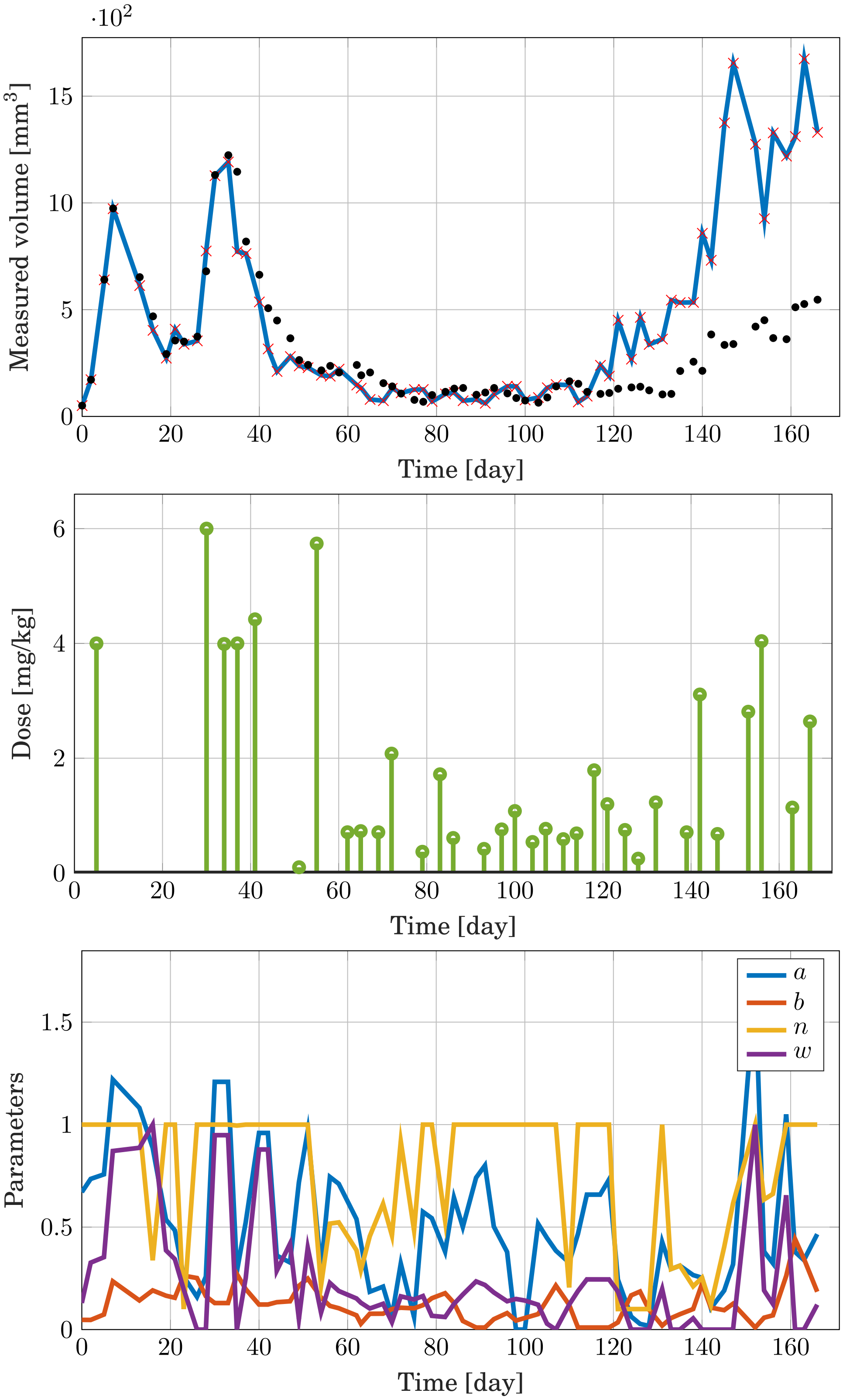


Figure 5.5: Evolution of the tumor volume, the corresponding doses, and the identified time-varying parameters for $S_{2,7}$. The red crosses represent the tumor volume measurements, while the black dots show their estimations, produced by the MHE.

The results can be divided into three classes, each with unique dynamic behavior. The algorithm worked as expected in the first case, as demonstrated in Figure 5.6. The magnitude of the doses is proportional to tumor volume, which is in strong contrast with the NMPC-MHE results in $\boldsymbol{S}_2$. Larger initial doses led to a strong remission in the subject, followed by subsequent lower maintenance doses until drug resistance appeared. Note the sudden increase in the computed doses at day 67 due to the change in the algorithm to weekly dosing. One can also see that the parameter variations are less abrupt than in the second experiment (c.f. Figure 5.4.). The same overall behavior can be seen in subjects $S_{3,1}$, $S_{3,2}$, $S_{3,3}$, $S_{3,4}$, $S_{3,6}$, $S_{3,8}$, $S_{3,9}$, $S_{3,12}$, $S_{3,13}$, $S_{3,15}$, $S_{3,16}$, and $S_{3,19}$. In the case of $S_{3,13}$, there was a significant measurement error on day 24, but it had no adverse impact on the calculated doses. Additionally, subject $S_{3,8}$ is currently still alive, with no palpable tumor.

In the second case, different dynamics can be observed. Here, initial doses do not lead to a longer remission period. Figure 5.8 shows the emergence of drug resistance immediately after the first remission. One can see how the initially applied doses are insufficient to significantly eradicate the tumor cells, leading to a fast adaptation to the drug by the tumor. This is also apparent in the case of $S_{3,7}$, $S_{3,10}$, and $S_{3,14}$. The same situation can occur if the initial dose is large, but the subsequent administrations are too small, which can be seen in the case of $S_{3,10}$, $S_{3,14}$, and $S_{3,21}$.

In the third case, there is only one a single subject, $S_{3,19}$, which can be seen in Figure 5.7. In this case, the magnitude of the subsequent doses was too large, leading to the immediate termination of the mouse on day 98 due to systematic toxicity. In experiment $\boldsymbol{S}_2$, the termination of subjects due to systematic toxicity was more prevalent, which indicates progress in the current control design. Nevertheless, the death of this mouse indicates that the constraint on the cumulated toxicity in the controller should be reworked in the following designs.

Some remark must be taken on the quality of the estimations. As one can see, the MHE could track the tumor volume evolution with minor errors, even in cases where resistance occurred as opposed to the second implementation. The estimated time-varying parameters also showed a decrease in variation. When the tumor volume stayed relatively constant, so did the parameters, and vice-versa, but nothing conclusive can be drawn from their individual values. In some instances, the parameter a, which captures the growth rate of the tumor, increases when resistance emerges ($S_{3,1}$,$S_{3,21}$), and in some instances, it declines ($S_{3,6}$, $S_{3,11}$).

Overall, the current experimental validation shows that the changes introduced in the algorithm lead to increased performance compared to previous implementations. Death caused by systematic toxicity is no longer prevalent among subjects. Additionally, the computed doses align with the measured tumor volumes, which can be greatly attributed to the lower variance in the estimated parameters.

5.5 Evaluation of the results

In order to verify the claims of the thesis group, a comparison must be made between the standard and algorithmic therapies. Both the survival times $t_{survival}$ and the total administered drug d_{sum} must be considered to evaluate the efficiency of the proposed algorithms. In practice, such a comparison is challenging to conduct due to many interplaying factors stemming from the biological nature of the experiments and the experimental setup. In this section, the subjects that received the MHE-NMPC algorithmic therapies in $\boldsymbol{S}_1$ are denoted by $\boldsymbol{S}_{1a}$, in $\boldsymbol{S}_2$ by $\boldsymbol{S}_{2a}$, and in $\boldsymbol{S}_3$ by $\boldsymbol{S}_{3a}$.

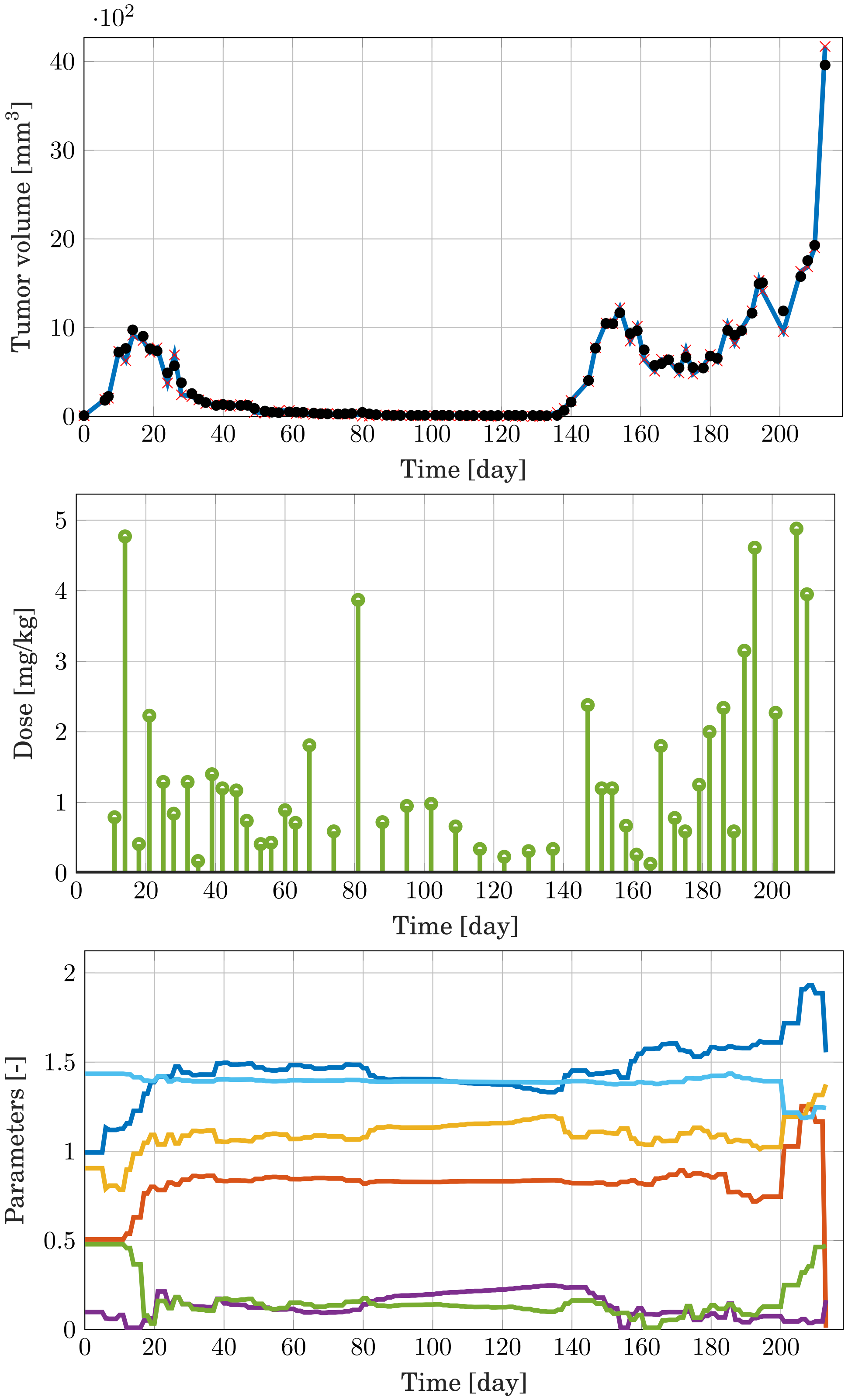


Figure 5.6: Evolution of the tumor volume, the corresponding doses, and the identified time-varying parameters for $S_{3,5}$. The red crosses represent the tumor volume measurements, while the black dots show their estimations produced by the MHE.

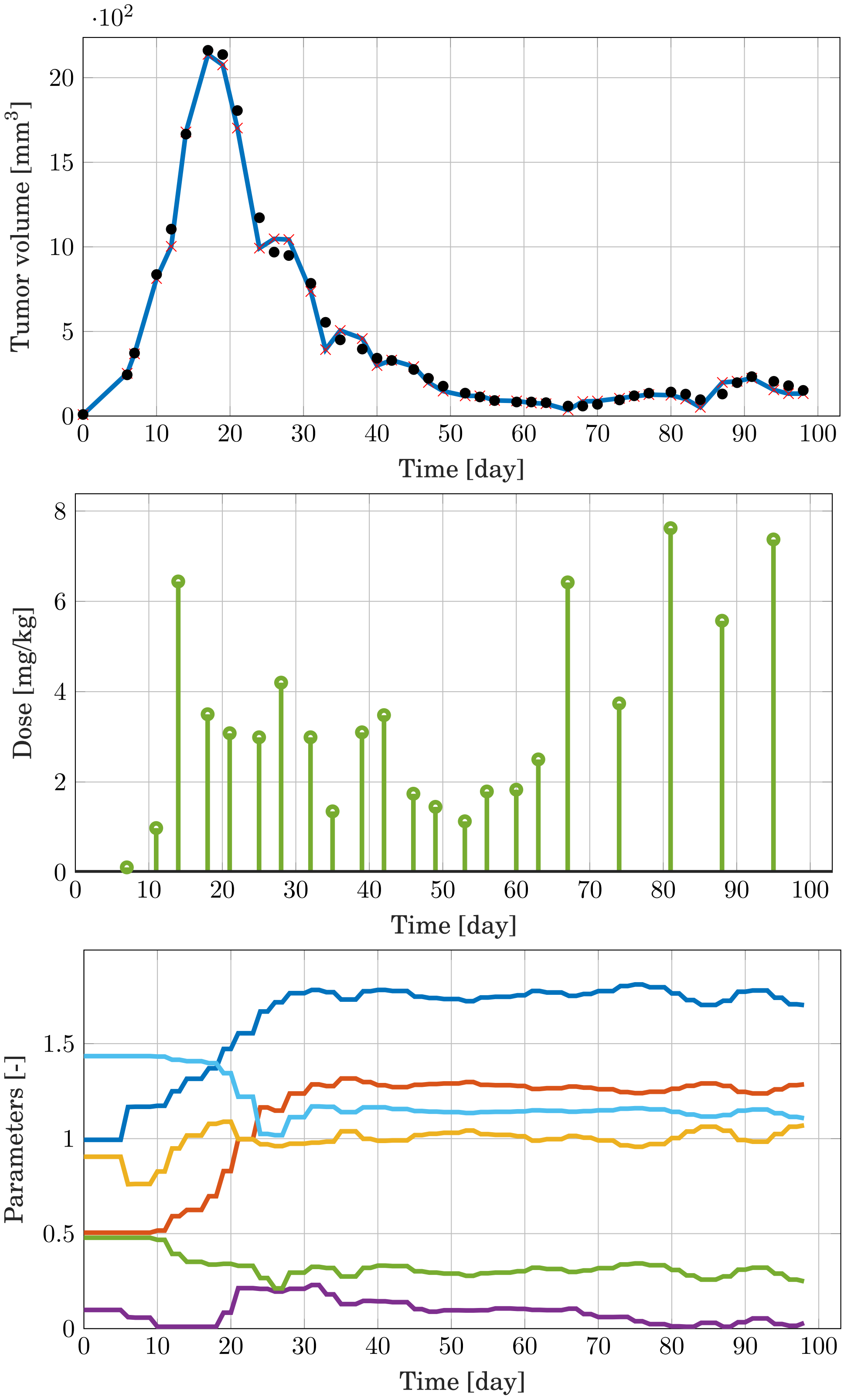


Figure 5.7: Evolution of the tumor volume, the corresponding doses, and the identified time-varying parameters for $S_{3,19}$. The red crosses represent the tumor volume measurements, while the black dots show their estimations produced by the MHE. For the parameters a, b, n, w, ED_{50} and c in the bottom figure, the colors blue, red, yellow, purple, green, and turquoise are assigned, respectively.

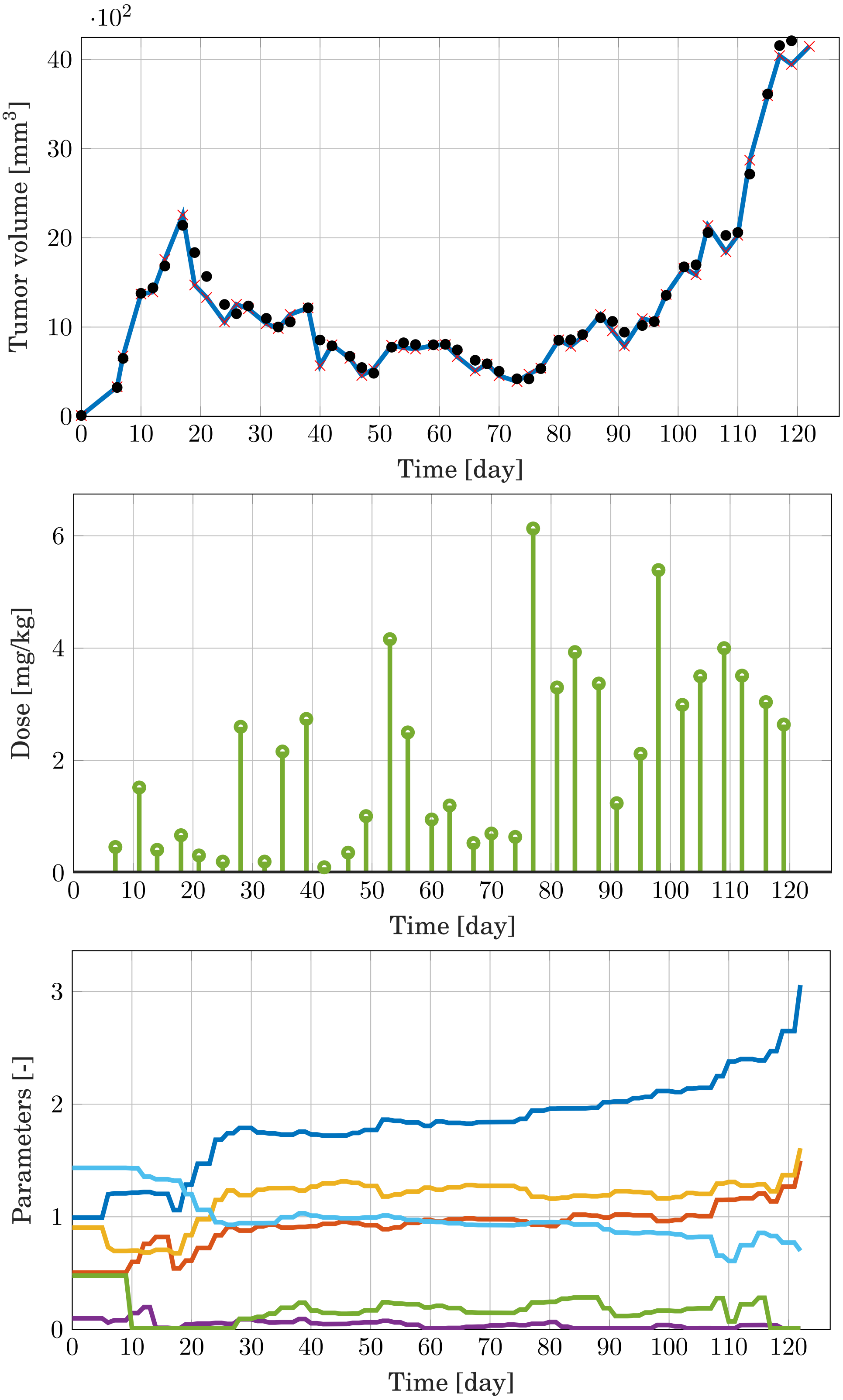


Figure 5.8: Evolution of the tumor volume, the corresponding doses, and the identified time-varying parameters for $S_{3,21}$. The red crosses represent the tumor volume measurements, while the black dots show their estimations produced by the MHE. For the parameters a, b, n, w, ED_{50} and c in the bottom figure, the colors blue, red, yellow, purple, green, and turquoise are assigned, respectively.

The first issue is that solely the survival times do not lead to a meaningful metric. The tumor grows at a different rate in each mouse, which means that the time between the implantation of the tumor and the first dose could be significant. Drug resistance causes another major issue. A simple metric would be to take $t_{survival}$ and d_{sum}, normalize them to the same range, average their values among the subjects in the experiment, and compare the final values across different groups. The problem is that different tumors react to the drug vastly differently. For example, comparing the diagrams B.4, B.5, B.6, B.7, B.8, B.9, B.10, and B.11 in Appendix B., one can see how the initial doses of 4 [mg/kg] lead to different time intervals between the time of injection and the time of relapse (when the tumor trigger becomes active again). A thorough understanding of drug resistance is required to compensate for the variation in drug efficiency among the subjects, which is currently unavailable.

Regarding the experiment setups, the following factors affect the comparison. First, the experimental protocols are different, thus, it is hard to compare them directly. For example, the NMPC-MHE therapy $\boldsymbol{S}_{1a}$ only lasted for a month, contrary to $\boldsymbol{S}_{2a}$ and $\boldsymbol{S}_{3a}$. Also, in $\boldsymbol{S}_{2a}$, the therapies had to be stopped earlier than the termination of mice due to ethical considerations arising from the toxicity levels of the mice. In $\boldsymbol{S}_{3a}$, weekly dosing was introduced from day 67 to protect the tail vein of the mice. There is also some variation in the applied doses between the baseline therapy $\boldsymbol{S}_0$ and the control group $\boldsymbol{S}_{2c}$. In $\boldsymbol{S}_0$ and $\boldsymbol{S}_{3c}$, the standard protocol was used, while in $\boldsymbol{S}_{2c}$ the doses were modified to 6 [mg/kg], and there was no control group in $\boldsymbol{S}_1$. Additionally, the length, non-repeatability, and the limited number of subjects in the experiments pose further difficulties in comparing the results.

The current survival times and the sum of doses can be seen in Table 5.2 for $\boldsymbol{S}_{1a}$, Table 5.3 for $\boldsymbol{S}_{2c}$, Table 5.4 for $\boldsymbol{S}_{2a}$, Table 5.5 for $\boldsymbol{S}_{3c}$, and finally Table 5.6 for $\boldsymbol{S}_{3a}$.

Due to the significant variations between the therapeutic protocols, and the biological nature of the experiments, two different comparisons were made regarding the third thesis group. The first analysis was performed between the experiments $\boldsymbol{S}_0$, $\boldsymbol{S}_{2c}$, and $\boldsymbol{S}_{2a}$, while the second involved $\boldsymbol{S}_0$, $\boldsymbol{S}_{3c}$, and $\boldsymbol{S}_{3a}$. These comparisons were performed using the Kaplan Meier (KM) estimator, which is a nonparametric estimator of the survival function of the subjects regarding an experiment [R66]. To compare the results of the algorithmic therapy in the second experiment with the baseline therapy, the KM curves were generated from the second dose of each experiment, excluding the effect of the identification dose. The reason behind this choice was to only compare the effect of the algorithmically generated doses on the survival times. In the case of the third experiment, the KM curves were generated from the first administration, since no identification dose was present in this case. The KM plots can be seen in Figures 5.9, and 5.10.

To statistically compare the KM curves, the logrank test was used [R67]. The logrank test is a hypothesis test to compare the survival distributions of two samples, denoted by $\text{lr}(\boldsymbol{S}_i, \boldsymbol{S}_j)$. The test provides a p value, which represents the likelihood that the two distributions are different. In particular, the lower the p value, the more likely the difference between two distributions can be attributed to the nature of the therapies rather than statistical fluctuations. In this dissertation, p values lower than 0.05 were considered acceptable.

First, the second experiment was compared with its control group, as $\text{lr}(\boldsymbol{S}_{2c}, \boldsymbol{S}_{2a})$. The logrank test resulted in $p = 0.0002$, which means that there is a high chance that the two distributions are fundamentally different. Second, the two control groups were compared as $\text{lr}(\boldsymbol{S}_0, \boldsymbol{S}_{2c})$, which resulted in $p = 0.5974$, showing no difference between the two control groups. Finally, the second experiment was compared with the baseline therapy as $\text{lr}(\boldsymbol{S}_0, \boldsymbol{S}_{2a})$. This led to $p = 0.0031$, which indicates a difference between the baseline and the algorithmic therapy.

Table 5.1: Logrank test p values for each experiment.

	$\mathrm{lr}(\boldsymbol{S}_{2c}, \boldsymbol{S}_{2a})$	$\mathrm{lr}(\boldsymbol{S}_0, \boldsymbol{S}_{2c})$	$\mathrm{lr}(\boldsymbol{S}_0, \boldsymbol{S}_{2a})$	$\mathrm{lr}(\boldsymbol{S}_{3c}, \boldsymbol{S}_{3a})$	$\mathrm{lr}(\boldsymbol{S}_0, \boldsymbol{S}_{3c})$	$\mathrm{lr}(\boldsymbol{S}_0, \boldsymbol{S}_{3a})$
p	0.0002	0.5974	0.0031	0.809	0.2052	0.1614

Regarding the third experiment, similar tests were made between the groups. The first comparison was made between the control group and the algorithmic therapy, i.e. $\mathrm{lr}(\boldsymbol{S}_{3c}, \boldsymbol{S}_{3a})$, which resulted in $p = 0.809$. This indicates that there is essentially no difference between the standard and the algorithmic therapy using the logrank test. The second comparison was between the baseline therapy and the control group as $\mathrm{lr}(\boldsymbol{S}_0, \boldsymbol{S}_{3c})$, which resulted in $p = 0.2052$. The last test was done between the baseline experiment and the algorithmic therapy as $\mathrm{lr}(\boldsymbol{S}_0, \boldsymbol{S}_{3a})$, where the result was $p = 0.1614$. While this p value is lower than the one obtained between the algorithmic therapy and the control group, it still indicates no difference between the therapies.

The logrank test showed that in the case of the MHE-NMPC approach in the second experiment, the algorithmic therapy can prolong the overall survival of mice. Contrary to this result, the algorithmic therapy in the third experiment has not performed satisfactory according to the logrank test. There are a number of reasons which might attribute to this result. As it was described in Section 5.4., small starting doses often led to drug resistance quickly. If the algorithm starts with a large initial dose each time, the survival of the mice could be improved. In the second experiment, 13 mice received starting doses between $5-6$ [mg/kg], while in the third experiment, only 7 mice received more than 5 [mg/kg], which could attribute to the short survival time of these subjects. Moreover, the introduction of the weekly dosing might have altered the effectiveness of the algorithmic therapy, which requires further investigation. Additionally, the subjects in the control group of the third experiment had more durable response to the standard therapy than the subjects in the baseline experiment. While the logrank test showed no difference between the two groups, the calculated value $p = 0.2052$ shows low similarity between the two survival curves.

The logrank test is also unable to account for long term survivals. In particular, subject $S_{3,8}$ is still living, with no palpable tumor. While this is only 4.76% of all the mice in the algorithmic group, it is still a significant achievement. The reason is that such a long overall survival has not been reported previously in the literature via conventional therapy using the same drug and targeting the same tumor variant. In a clinical setting, even the slightest chance of durable survival is considerable. Additionally, this is achieved by applying lower doses, which mitigates the side effects during the treatment, thus, improving the quality of life of the patient. In summary, even if the logrank test do not show any significant difference between the two groups, the lower side effects and the small chance of a durable survival could make the algorithm applicable with small adjustments regarding the magnitude of the initial dose.

The results confirm the claims of the thesis group. The first iteration of the algorithm was indeed able to induce remission of the subjects, and the amount of the administered dose was in the same range as it would be in the case of the standard protocol. In the case of the second implementation, the algorithm outperformed both the control group and the baseline therapy. Nevertheless, additional improvements must be made in the future so that the early onset of drug resistance can be avoided entirely, as could be seen in the third experiment.

Table 5.2: Sum of doses [mg/kg] and survival times [day] of the MHE-NMPC algorithm treated subjects in the first experiment.

	$S_{1,1}$	$S_{1,2}$	$S_{1,3}$	$S_{1,5}$	$S_{1,6}$
d_{sum}	14.2	10.73	12.27	13.18	17.36
$t_{survival}$	102	112	112	165	93

Table 5.3: Sum of doses [mg/kg] and survival times [day] of the subjects in the control group of the second experiment.

	$S_{2c,1}$	$S_{2c,2}$	$S_{2c,3}$	$S_{2c,4}$	$S_{2c,5}$	$S_{2c,6}$	$S_{2c,7}$
d_{sum}	36	30	6	30	36	24	36
$t_{survival}$	93	154	89	196	154	112	152

Table 5.4: Sum of doses [mg/kg] and survival times [day] of the MHE-NMPC algorithm treated subjects in the second experiment.

	$S_{2,1}$	$S_{2,2}$	$S_{2,3}$	$S_{2,4}$	$S_{2,5}$	$S_{2,6}$	$S_{2,7}$
d_{sum}	70.09	63.41	83.91	76.99	75.01	73.88	57.74
$t_{survival}$	131	166	124	191	198	194	166
	$S_{2,8}$	$S_{2,9}$	$S_{2,10}$	$S_{2,11}$	$S_{2,12}$	$S_{2,13}$	$S_{2,14}$
d_{sum}	66.51	48.37	93.32	108.98	73.75	80.85	71.44
$t_{survival}$	159	89	341	313	348	223	191
	$S_{2,15}$	$S_{2,16}$	$S_{2,17}$	$S_{2,18}$	$S_{2,19}$	$S_{2,20}$	$S_{2,21}$
d_{sum}	95.8	77	88.36	51.03	50.08	85.68	81.68
$t_{survival}$	292	168	168	82	159	229	348

Table 5.5: Sum of doses [mg/kg] and survival times [day] of the subjects in the control group of the third experiment.

	$S_{3c,1}$	$S_{3c,2}$	$S_{3c,3}$	$S_{3c,4}$	$S_{3c,5}$	$S_{3c,6}$
d_{sum}	48	40	48	48	48	48
$t_{survival}$	231	185	306	220	220	234
	$S_{3c,7}$	$S_{3c,8}$	$S_{3c,9}$	$S_{3c,10}$	$S_{3c,11}$	$S_{3c,12}$
d_{sum}	24	24	48	32	32	48
$t_{survival}$	82	91	262	115	133	234

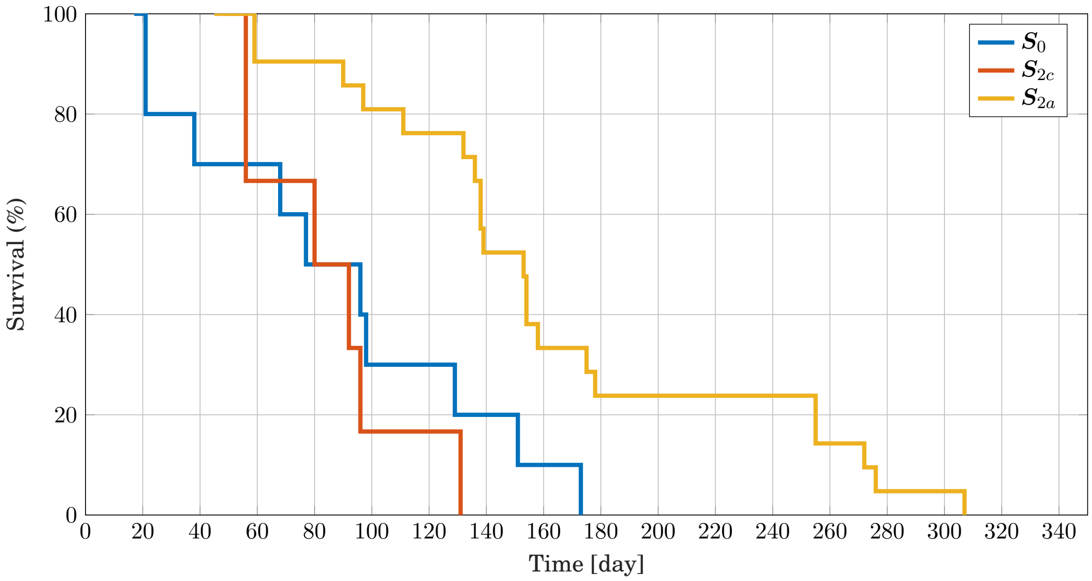


Figure 5.9: Kaplan Meier curves, associated with the first comparison, where the curves were generated assuming $t = 0$ to be the day of the second administration for each subject. The blue curve corresponds to the baseline experiment $\boldsymbol{S}_0$, the red curve to the control group $\boldsymbol{S}_{2c}$, and the yellow curve to the MHE-NMPC algorithm in the second experiment $\boldsymbol{S}_{2a}$.

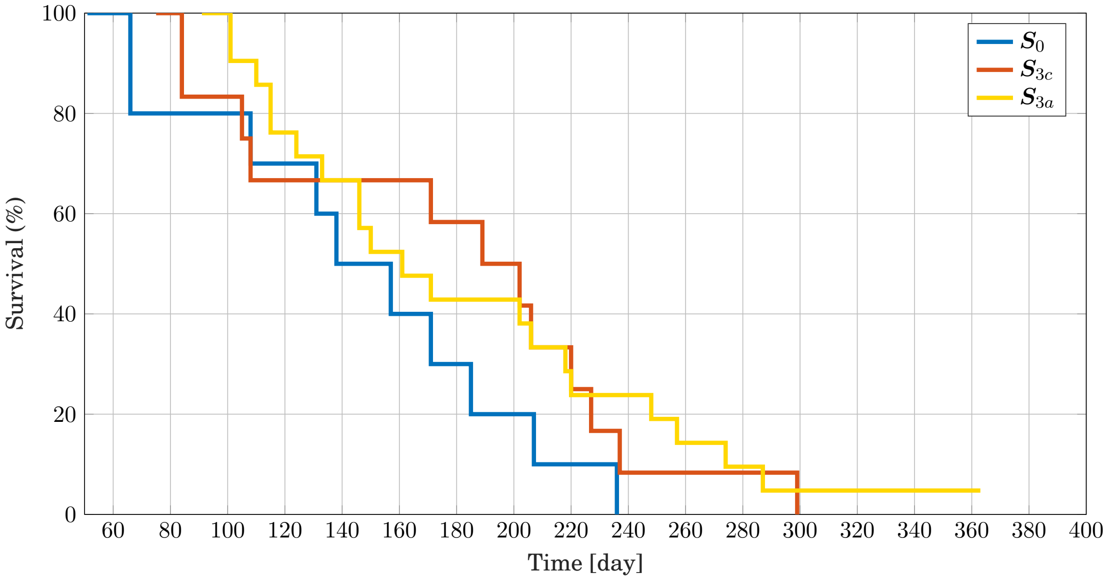


Figure 5.10: Kaplan Meier curves, associated with the second comparison, where the curves were generated assuming $t = 0$ to be the day of the first administration for each subject. The blue curve corresponds to the baseline experiment $\boldsymbol{S}_0$, the red curve to the control group $\boldsymbol{S}_{3c}$, and the yellow curve to the MHE-NMPC algorithm in the third experiment $\boldsymbol{S}_{3a}$. The end of the yellow curve represents subject $S_{3,8}$, which is currently still alive.

Table 5.6: Sum of doses [mg/kg] and survival times [day] of the MHE-NMPC algorithm treated subjects in the third experiment, up to day 386.

	$S_{3,1}$	$S_{3,2}$	$S_{3,3}$	$S_{3,4}$	$S_{3,5}$	$S_{3,6}$	$S_{3,7}$
d_{sum}	67.92	70.2	63.94	70.6	63.59	32.78	62.55
$t_{survival}$	157	271	157	229	213	185	108
	$S_{3,8}$	$S_{3,9}$	$S_{3,10}$	$S_{3,11}$	$S_{3,12}$	$S_{3,13}$	$S_{3,14}$
d_{sum}	21.35	46.76	72.02	87.85	36.56	40.84	72.04
$t_{survival}$	-	157	122	255	213	227	131
	$S_{3,15}$	$S_{3,16}$	$S_{3,17}$	$S_{3,18}$	$S_{3,19}$	$S_{3,20}$	$S_{3,21}$
d_{sum}	85.84	106.42	70.57	50.87	73.38	71.89	68.58
$t_{survival}$	168	285	294	117	98	140	122

6 Conclusions

In my dissertation, I described the development of a fully automated tumor regulation algorithm on the basis of different control techniques. I have investigated the use of the RFPT method, for which I have developed different strategies based on the IO linearization principle. While the method has promising applications in other technical domains, unfortunately, it did not bring the expected performance benefits for the tumor growth regulation problem.

As a consequence, an NMPC was designed through multiple implementations in conjunction with an MHE. In order to properly tune the MHE, I have implemented the corresponding FIE, which was able to provide reliable initial parameter estimates for the SAEM method. Using the MHE, I have also introduced time-varying model parameters, which has significantly improved the prediction performance of the model. In addition, a special emphasis were taken on constraining the parameter values, such that they describe physiologically feasible tumor behavior.

To generate the optimal administration protocols, several NMPC implementations were carried out. In the first implementation the DMS approach was used, which was then tested both in silico and in vivo. The algorithm had to be tuned manually for each identified parameter set, which was a major drawback during the in vivo validation. This issue was solved in the second implementation, where an automated tuning procedure was introduced and a number of modifications were applied on the optimization problem. The experimental validation of the second controller showed, that while it was successful in extending the overall survival of the group, the consistency of the applied dosages was unsatisfactory. The issues present in the second experiment then were solved in the third implementation of the controller, where the MHE was completely redesigned to eliminate the large variation between the computed time-varying model parameters. Modifications were also introduced in the NMPC design, and the combined approach is currently tested in silico. The obtained result at this point indicate, that the current design was able to solve all the previous problems and can operate in a fully automated manner.

There is a large number of possible research directions in the future. The first and foremost important is the investigation of possible application of the algorithm in clinical use. Since caliper measurements can not be used in human trials due to the location of the tumor, alternative measurement devices must be taken into account. The majority of imaging techniques are expensive and time consuming to be used frequently, additional research should be conducted where the measurements are taken weekly or even more sparsely. A possible adaptation strategy in human use would be the optimization of a regular course of therapy. This means that the medical professionals determine the measurement and administration time instances, as in the case of a conventional chemotherapy, for which the algorithm computes the state estimates and the optimized doses. In the near future, additional measurement channels might be available, which can significantly improve the estimation of the model parameters. In particular, measuring the concentration of drug in the blood during the experiment without sacrificing the animal would greatly improve the estimation of the pharmacokinetic parameters in the model. Measuring different tumor markers would also be beneficial because they allow more frequent tumor size measurement. Furthermore, different tumor variants and chemotherapeutic agents should also be tested. In principle, the flexibility of the model and the handling of time-varying parameters should permit the algorithm to be used for the treatment of different tumors by tailoring the model parameters to these cases.

Appendix A

Discrete time terms

$$
\begin{aligned}
h^r(\boldsymbol{x},u) = {} & (-(-1+\Delta t w)^3)x_2 + (x_1((-a)b^2c\Delta t^4x_3^2 - b^2c\Delta t^4wx_3^2 + abc\Delta t^3x_3(ED_{50}+x_3) \\
& + a^2bc\Delta t^4x_3(ED_{50}+x_3) - abc\Delta t^4nx_3(ED_{50}+x_3) + bc\Delta t^3wx_3(ED_{50}+x_3) \\
& + abc\Delta t^4wx_3(ED_{50}+x_3) - bc\Delta t^4nwx_3(ED_{50}+x_3) + ab^2\Delta t^4ux_3(K_b+x_3) \\
& + b^2\Delta t^4uwx_3(K_b+x_3) + ab^2\Delta t^3x_3^2(K_b+x_3) + b^2\Delta t^3wx_3^2(K_b+x_3) \\
& - ab\Delta t^3u(ED_{50}+x_3)(K_b+x_3) - a^2b\Delta t^4u(ED_{50}+x_3)(K_b+x_3) \\
& + ab\Delta t^4nu(ED_{50}+x_3)(K_b+x_3) - b\Delta t^3uw(ED_{50}+x_3)(K_b+x_3) \\
& - ab\Delta t^4uw(ED_{50}+x_3)(K_b+x_3) + b\Delta t^4nuw(ED_{50}+x_3)(K_b+x_3) \\
& - ab\Delta t^2x_3(ED_{50}+x_3)(K_b+x_3) - a^2b\Delta t^3x_3(ED_{50}+x_3)(K_b+x_3) \\
& + ab\Delta t^3nx_3(ED_{50}+x_3)(K_b+x_3) - b\Delta t^2wx_3(ED_{50}+x_3)(K_b+x_3) \\
& - ab\Delta t^3wx_3(ED_{50}+x_3)(K_b+x_3) + b\Delta t^3nwx_3(ED_{50}+x_3)(K_b+x_3) \\
& - 2ab\Delta t^2x_3((-c)\Delta tx_3 + ED_{50}(K_b+x_3) + \Delta tu(K_b+x_3) + x_3(K_b+x_3)) \\
& - a^2b\Delta t^3x_3((-c)\Delta tx_3 + ED_{50}(K_b+x_3) + \Delta tu(K_b+x_3) + x_3(K_b+x_3)) \\
& + ab\Delta t^3nx_3((-c)\Delta tx_3 + ED_{50}(K_b+x_3) + \Delta tu(K_b+x_3) + x_3(K_b+x_3)) \\
& - 2b\Delta t^2wx_3((-c)\Delta tx_3 + ED_{50}(K_b+x_3) + \Delta tu(K_b+x_3) + x_3(K_b+x_3)) \\
& + b\Delta t^3nwx_3((-c)\Delta tx_3 + ED_{50}(K_b+x_3) + \Delta tu(K_b+x_3) + x_3(K_b+x_3)) \\
& + b\Delta t^3w^2x_3((-c)\Delta tx_3 + ED_{50}(K_b+x_3) + \Delta tu(K_b+x_3) + x_3(K_b+x_3)) \\
& + (ED_{50}+x_3)((-c)\Delta tx_3 + ED_{50}(K_b+x_3) + \Delta tu(K_b+x_3) + x_3(K_b+x_3)) \\
& + 3a\Delta t(ED_{50}+x_3)((-c)\Delta tx_3 + ED_{50}(K_b+x_3) + \Delta tu(K_b+x_3) + x_3(K_b+x_3)) \\
& + 3a^2\Delta t^2(ED_{50}+x_3)((-c)\Delta tx_3 + ED_{50}(K_b+x_3) + \Delta tu(K_b+x_3) + x_3(K_b+x_3)) \\
& + a^3\Delta t^3(ED_{50}+x_3)((-c)\Delta tx_3 + ED_{50}(K_b+x_3) + \Delta tu(K_b+x_3) + x_3(K_b+x_3)) \\
& - 3a\Delta t^2n(ED_{50}+x_3)((-c)\Delta tx_3 + ED_{50}(K_b+x_3) + \Delta tu(K_b+x_3) + x_3(K_b+x_3)) \\
& - 2a^2\Delta t^3n(ED_{50}+x_3)((-c)\Delta tx_3 + ED_{50}(K_b+x_3) + \Delta tu(K_b+x_3) + x_3(K_b+x_3)) \\
& + a\Delta t^3n^2(ED_{50}+x_3)((-c)\Delta tx_3 + ED_{50}(K_b+x_3) + \Delta tu(K_b+x_3) + x_3(K_b+x_3)) \\
& - 3\Delta t^2nw(ED_{50}+x_3)((-c)\Delta tx_3 + ED_{50}(K_b+x_3) + \Delta tu(K_b+x_3) + x_3(K_b+x_3)) \\
& - a\Delta t^3nw(ED_{50}+x_3)((-c)\Delta tx_3 + ED_{50}(K_b+x_3) + \Delta tu(K_b+x_3) + x_3(K_b+x_3)) \\
& + \Delta t^3n^2w(ED_{50}+x_3)((-c)\Delta tx_3 + ED_{50}(K_b+x_3) + \Delta tu(K_b+x_3) + x_3(K_b+x_3)) \\
& + \Delta t^3nw^2(ED_{50}+x_3)((-c)\Delta tx_3 + ED_{50}(K_b+x_3) + \Delta tu(K_b+x_3) + x_3(K_b+x_3)))) \\
& /((ED_{50}+x_3)((-c)\Delta tx_3 + ED_{50}(K_b+x_3) + \Delta tu(K_b+x_3) + x_3(K_b+x_3))),
\end{aligned}
\tag{A.1}
$$

$$\begin{aligned}
u = \Psi(x, y(k+r)) = &-(((-v + (1 + a^3\Delta t^3 + a^2\Delta t^2(3 - 2b\Delta t - 2\Delta tn) + b^2\Delta t^3 w \\
&- 3\Delta t^2 nw + \Delta t^3 n^2 w + \Delta t^3 nw^2 + b\Delta t^2 w(-3 + \Delta t(2n + w)) + a\Delta t(3 + b^2\Delta t^2 \\
&- 3\Delta tn + \Delta t^2 n(n - w) + b\Delta t(-3 + 2\Delta tn - \Delta tw)))x_1 - (-1 + \Delta tw)^3 x_2)(c\Delta t \\
&- Kb - x_3)x_3^2 + ED_{50}^2(v - (1 + a^3\Delta t^3 + a^2\Delta t^2(3 - 2\Delta tn) + a\Delta t(3 - 3\Delta tn \\
&+ \Delta t^2 n(n - w)) - 3\Delta t^2 nw + \Delta t^3 nw(n + w))x_1 + (-1 + \Delta tw)^3 x_2)(Kb + x_3) \\
&+ ED_{50}x_3(c\Delta t(-v + (1 + a^3\Delta t^3 - a^2\Delta t^2(-3 + b\Delta t + 2\Delta tn) + a\Delta t(3 - 3\Delta tn \\
&+ b\Delta t(-1 + \Delta t(n - w)) + \Delta t^2 n(n - w)) - b\Delta t^2 w - 3\Delta t^2 nw + b\Delta t^3 nw + \Delta t^3 n^2 w \\
&+ \Delta t^3 nw^2)x_1 - (-1 + \Delta tw)^3 x_2) + (2v - (2 + 2a^3\Delta t^3 - 2a^2\Delta t^2(-3 + b\Delta t + 2\Delta tn) \\
&- 6\Delta t^2 nw + 2\Delta t^3 nw(n + w) + b\Delta t^2 w(-3 + \Delta t(2n + w)) + a\Delta t(6 - 6\Delta tn \\
&+ 2\Delta t^2 n(n - w) + b\Delta t(-3 + 2\Delta tn - \Delta tw)))x_1 + 2(-1 + \Delta tw)^3 x_2)(Kb + x_3))) \\
&/(\Delta t(Kb + x_3)(ED_{50}(v - (1 + a^3\Delta t^3 - a^2\Delta t^2(-3 + b\Delta t + 2\Delta tn) + a\Delta t(3 - 3\Delta tn \\
&+ b\Delta t(-1 + \Delta t(n - w)) + \Delta t^2 n(n - w)) - b\Delta t^2 w - 3\Delta t^2 nw + b\Delta t^3 nw \\
&+ \Delta t^3 n^2 w + \Delta t^3 nw^2)x_1 + (-1 + \Delta tw)^3 x_2) + (v - (1 + a^3\Delta t^3 + a^2\Delta t^2(3 - 2b\Delta t \\
&- 2\Delta tn) + b^2\Delta t^3 w - 3\Delta t^2 nw + \Delta t^3 n^2 w + \Delta t^3 nw^2 + b\Delta t^2 w(-3 + \Delta t(2n + w)) \\
&+ a\Delta t(3 + b^2\Delta t^2 - 3\Delta tn + \Delta t^2 n(n - w) + b\Delta t(-3 + 2\Delta tn - \Delta tw)))x_1 \\
&+ (-1 + \Delta tw)^3 x_2)x_3)))
\end{aligned} \tag{A.2}$$

Appendix B

Experimental diagrams

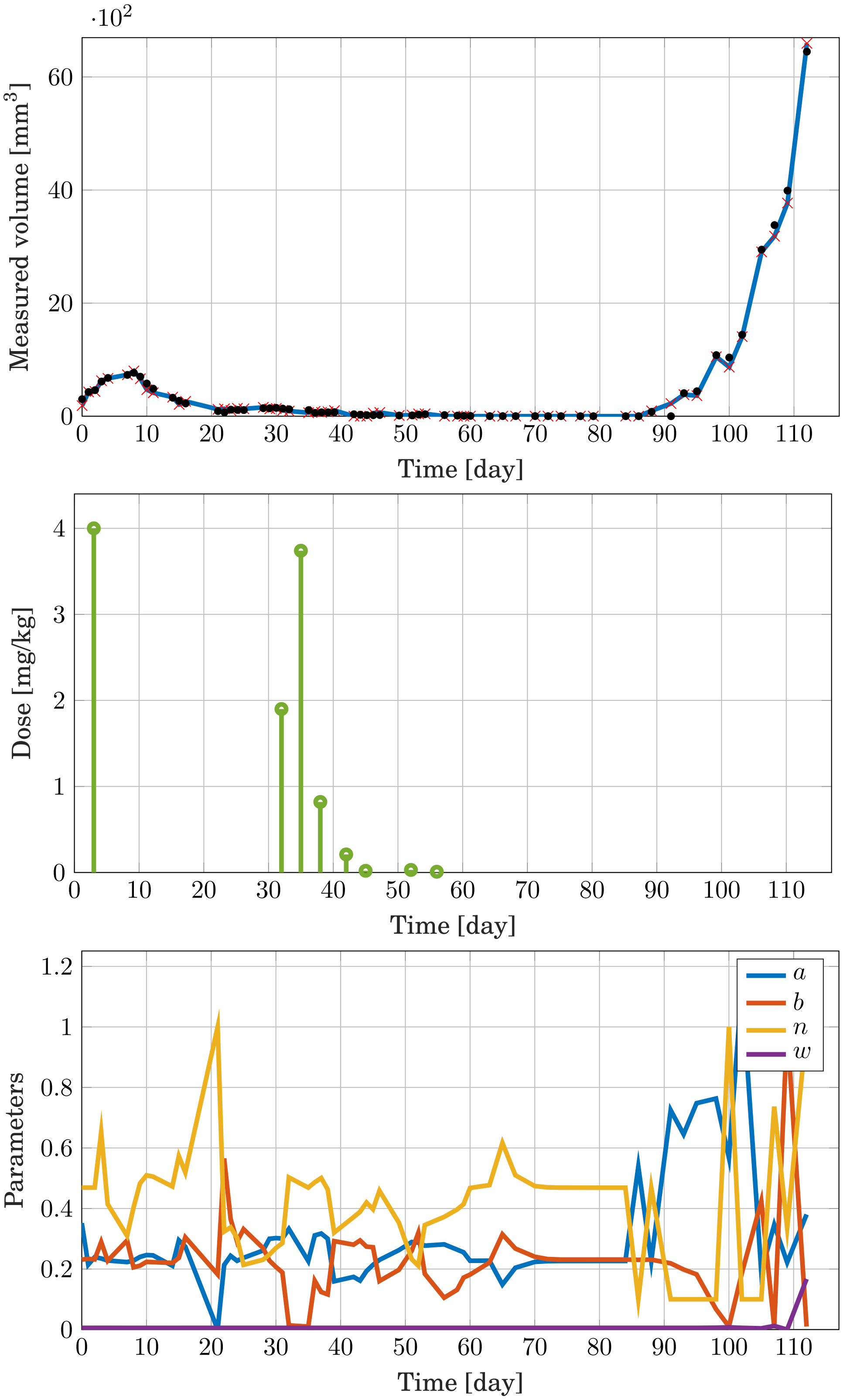


Figure B.1: Evolution of the tumor volume, the corresponding doses and the identified time-varying parameters for $S_{1,2}$. The red crosses represent the tumor volume measurements, while the black dots show their estimations, produced by the MHE.

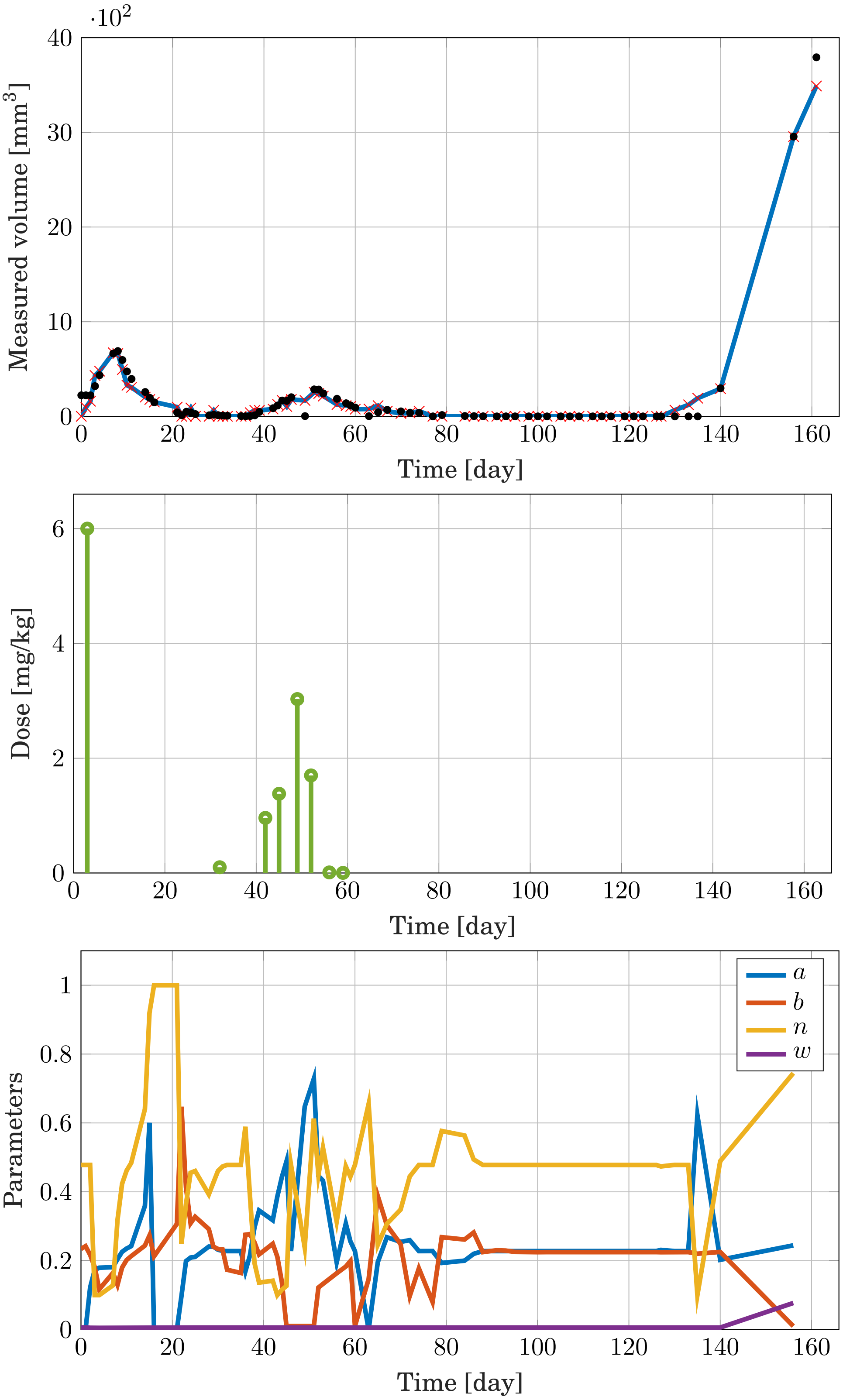


Figure B.2: Evolution of the tumor volume, the corresponding doses and the identified time-varying parameters for $S_{1,5}$. The red crosses represent the tumor volume measurements, while the black dots show their estimations, produced by the MHE.

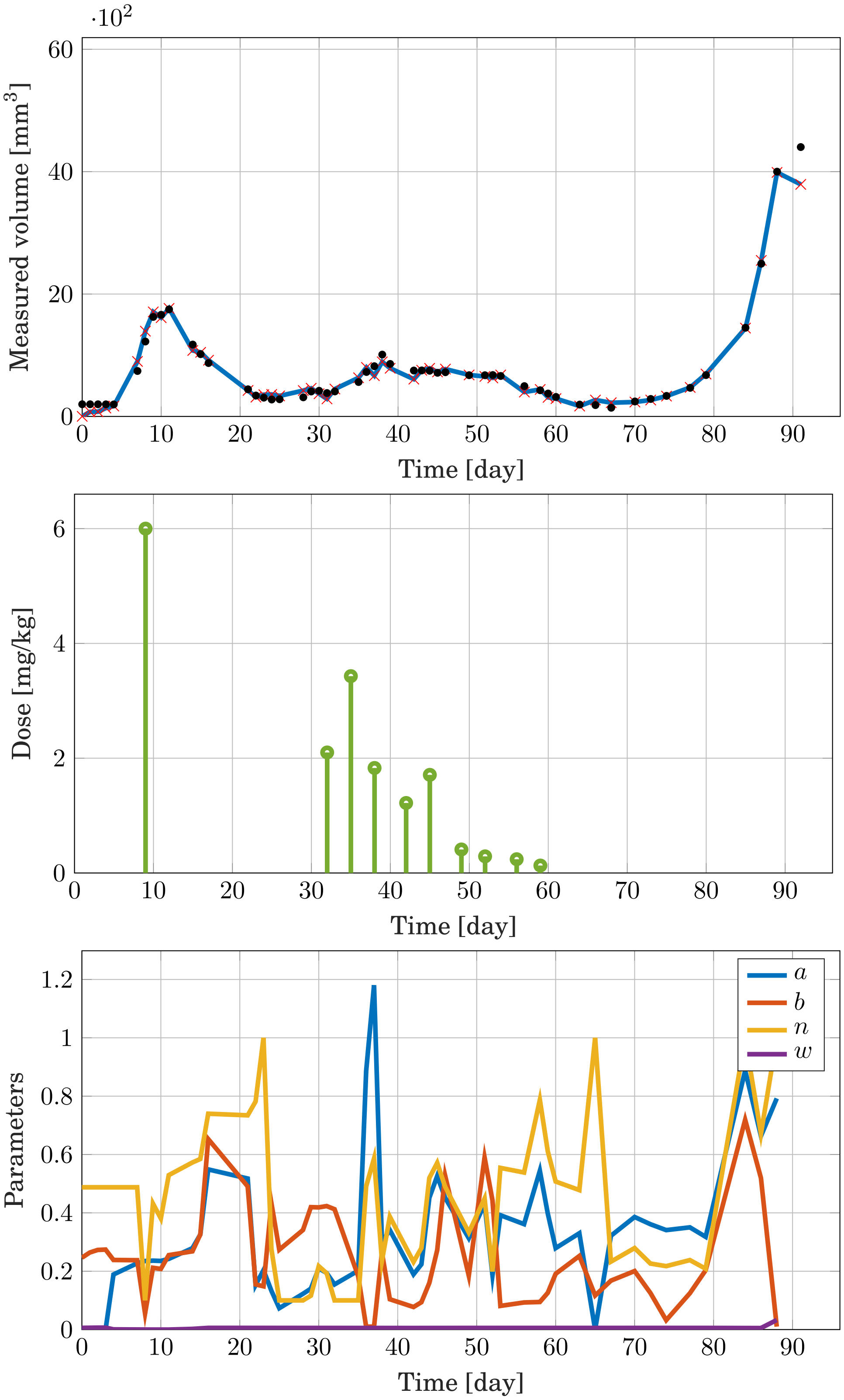


Figure B.3: Evolution of the tumor volume, the corresponding doses and the identified time-varying parameters for $S_{1,6}$. The red crosses represent the tumor volume measurements, while the black dots show their estimations, produced by the MHE.

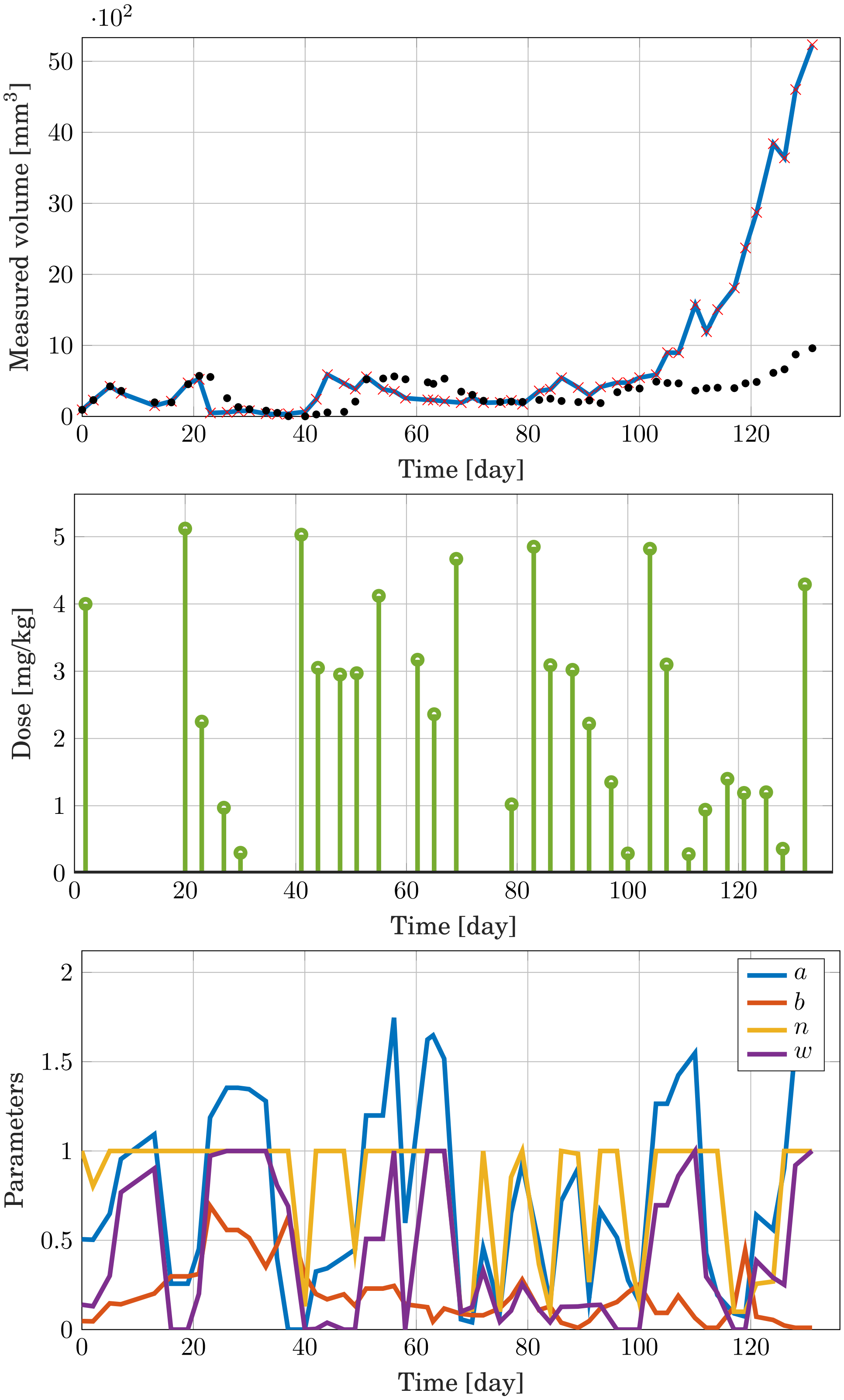


Figure B.4: Evolution of the tumor volume, the corresponding doses and the identified time-varying parameters for $S_{2,1}$. The red crosses represent the tumor volume measurements, while the black dots show their estimations, produced by the MHE.

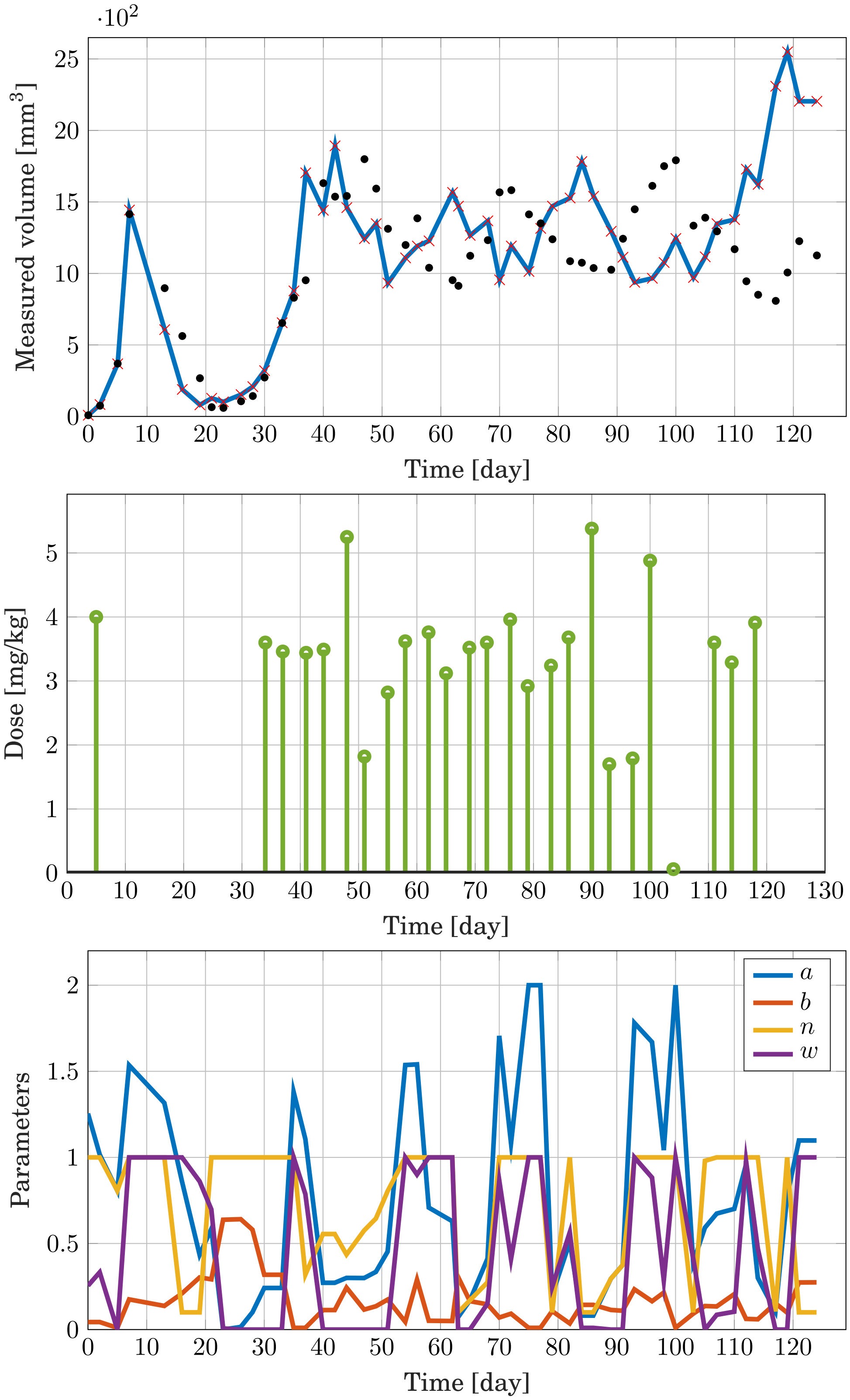


Figure B.5: Evolution of the tumor volume, the corresponding doses and the identified time-varying parameters for $S_{2,3}$. The red crosses represent the tumor volume measurements, while the black dots show their estimations, produced by the MHE.

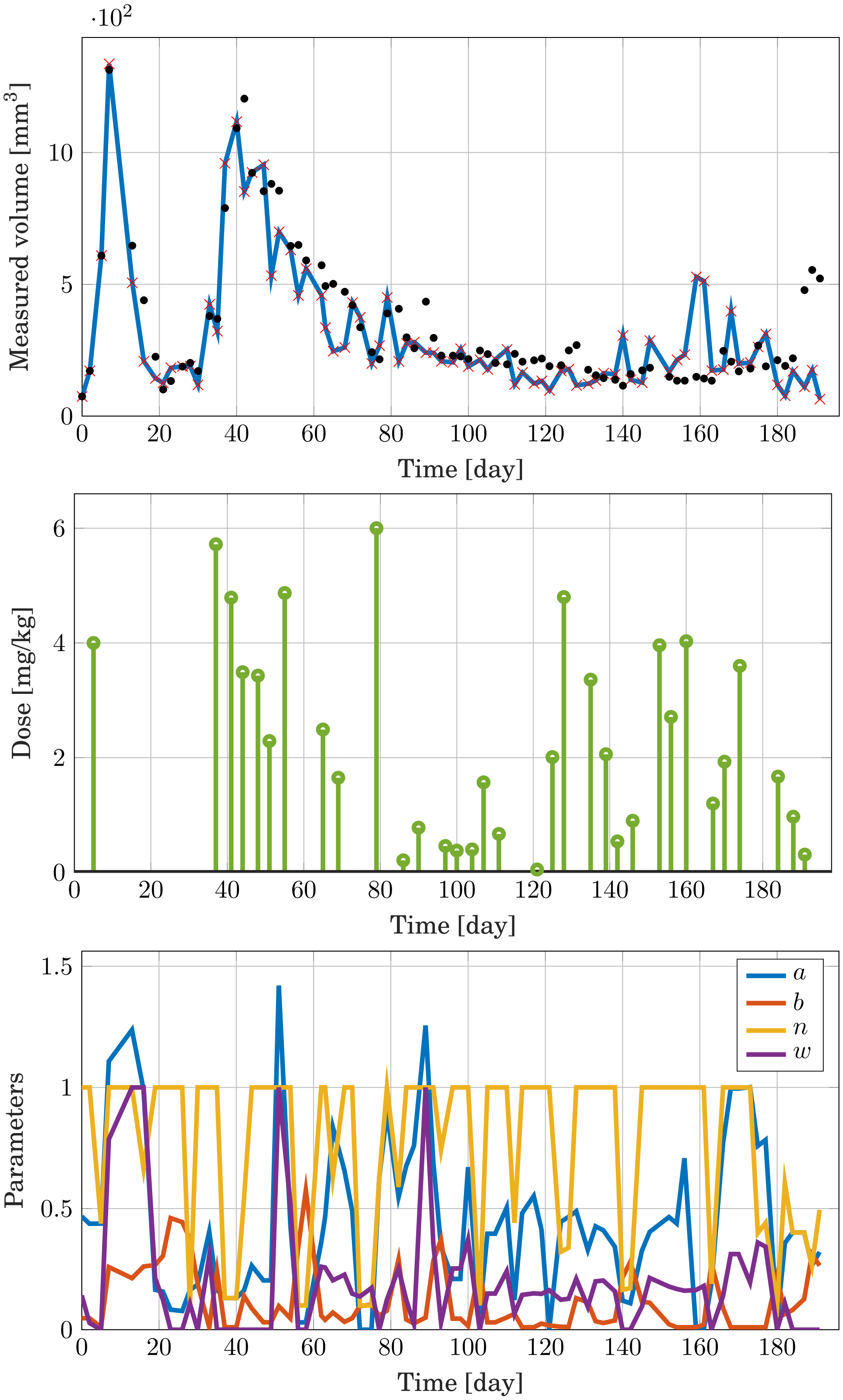


Figure B.6: Evolution of the tumor volume, the corresponding doses and the identified time-varying parameters for $S_{2,4}$. The red crosses represent the tumor volume measurements, while the black dots show their estimations, produced by the MHE.

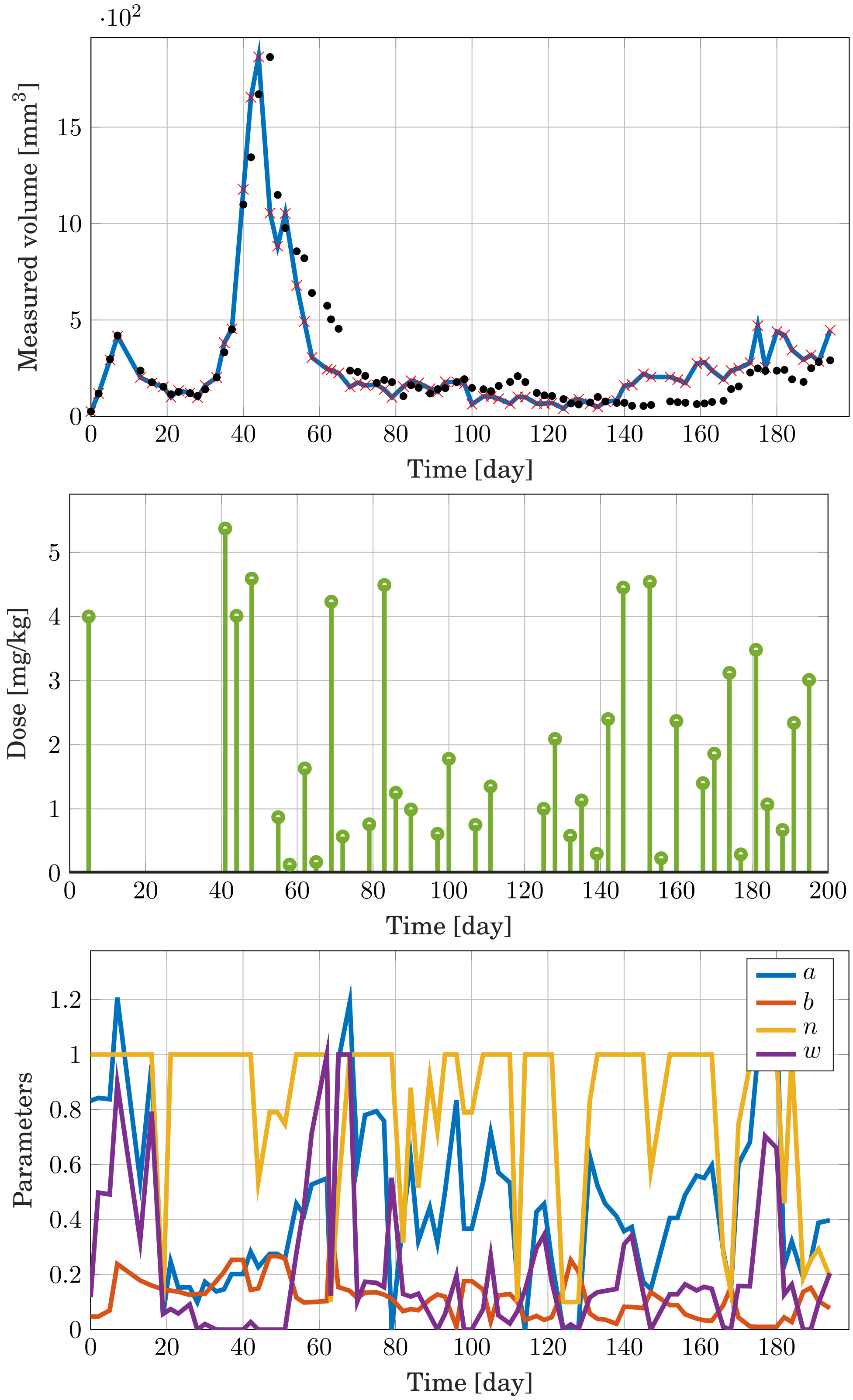


Figure B.7: Evolution of the tumor volume, the corresponding doses and the identified time-varying parameters for $S_{2,6}$. The red crosses represent the tumor volume measurements, while the black dots show their estimations, produced by the MHE.

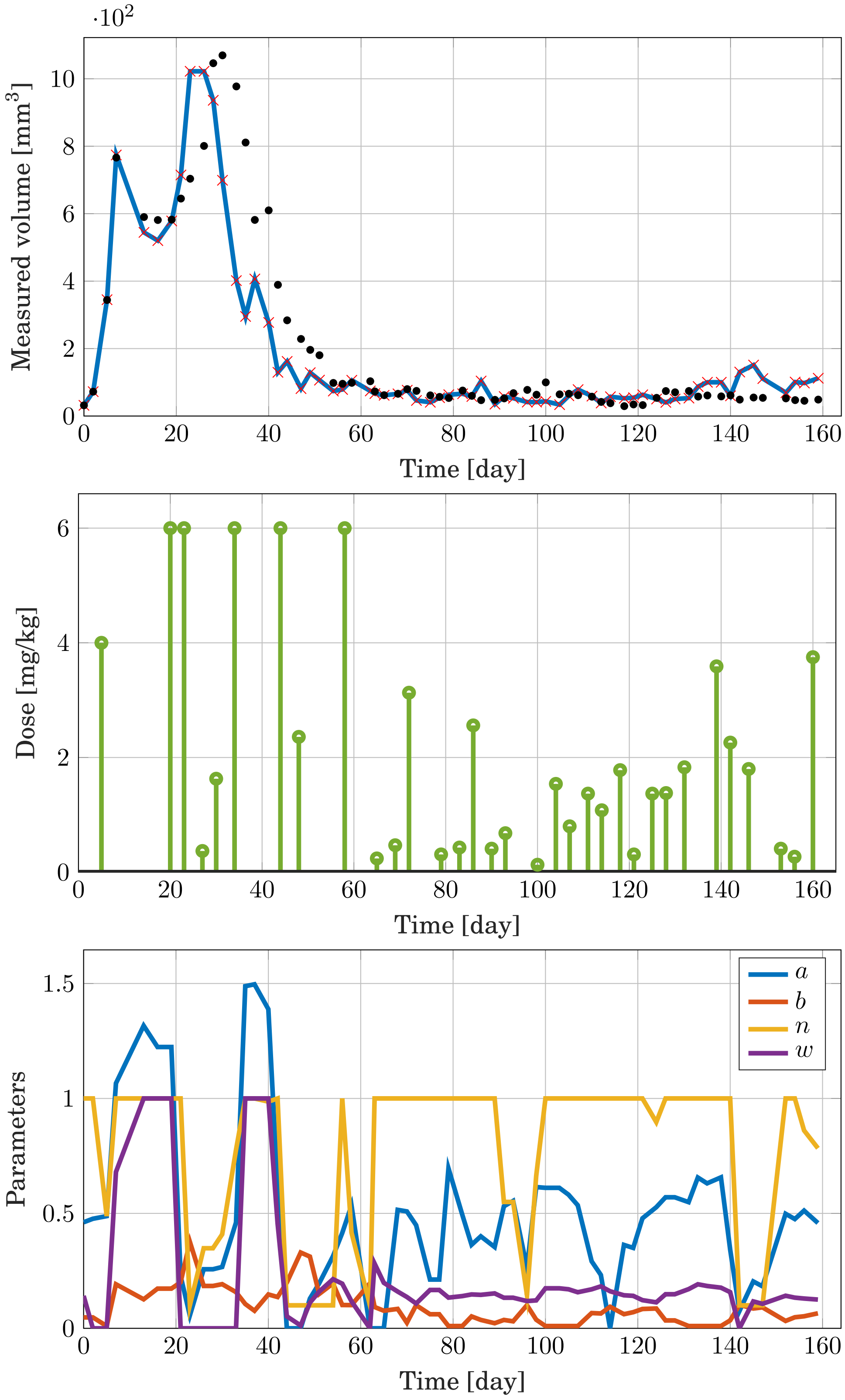


Figure B.8: Evolution of the tumor volume, the corresponding doses and the identified time-varying parameters for $S_{2,8}$. The red crosses represent the tumor volume measurements, while the black dots show their estimations, produced by the MHE.

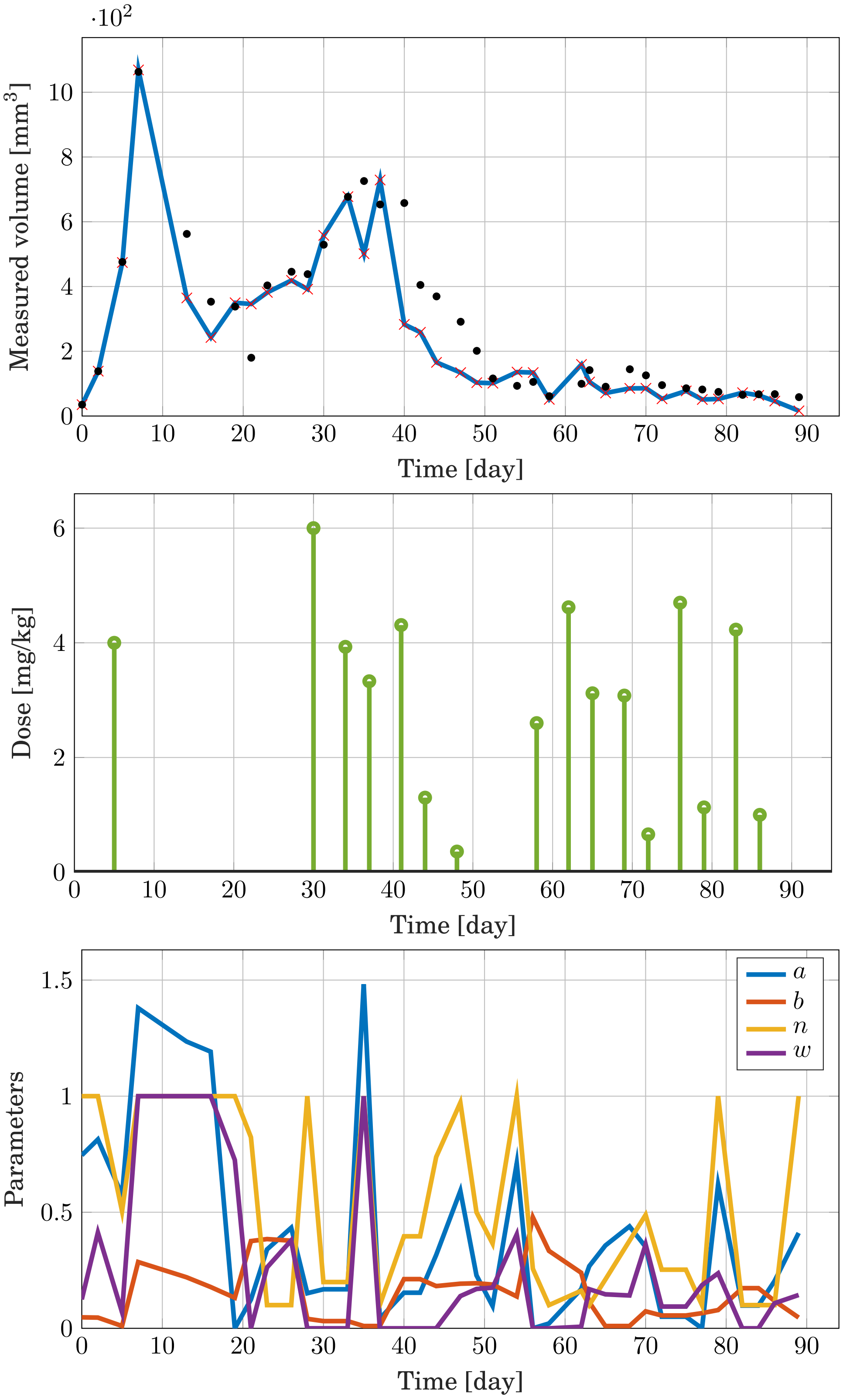


Figure B.9: Evolution of the tumor volume, the corresponding doses and the identified time-varying parameters for $S_{2,9}$. The red crosses represent the tumor volume measurements, while the black dots show their estimations, produced by the MHE.

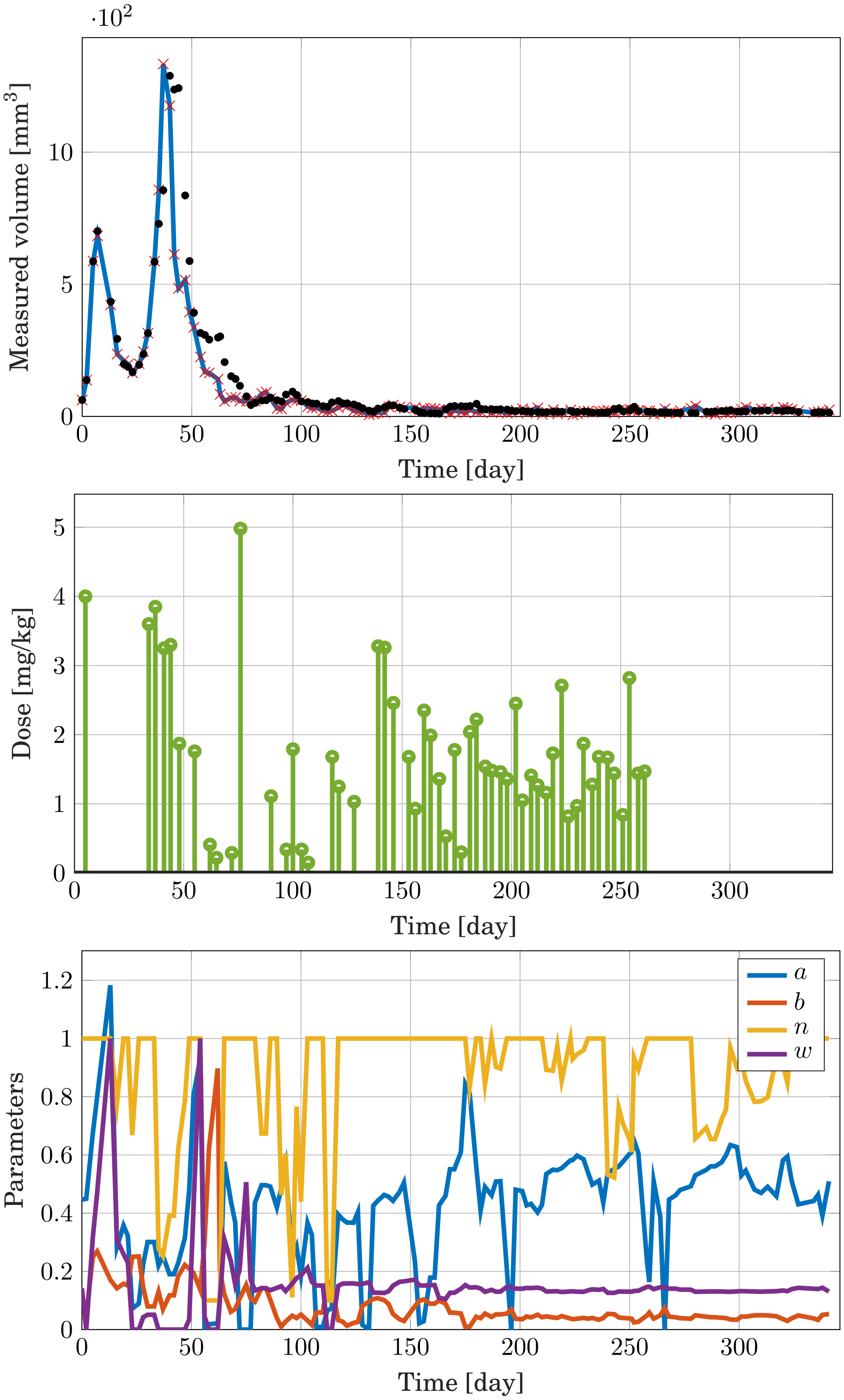


Figure B.10: Evolution of the tumor volume, the corresponding doses and the identified time-varying parameters for $S_{2,10}$. The red crosses represent the tumor volume measurements, while the black dots show their estimations, produced by the MHE.

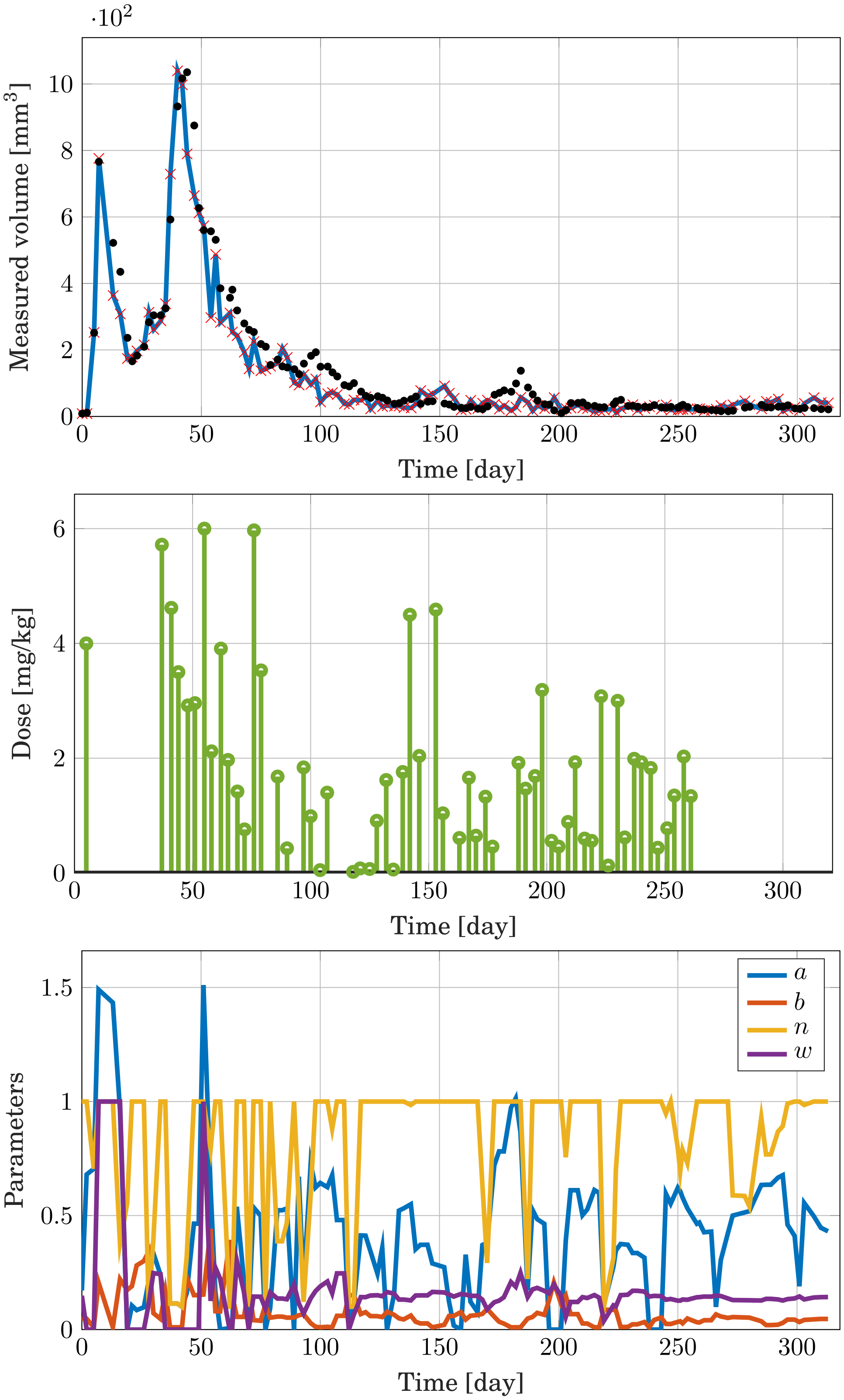


Figure B.11: Evolution of the tumor volume, the corresponding doses and the identified time-varying parameters for $S_{2,11}$. The red crosses represent the tumor volume measurements, while the black dots show their estimations, produced by the MHE.

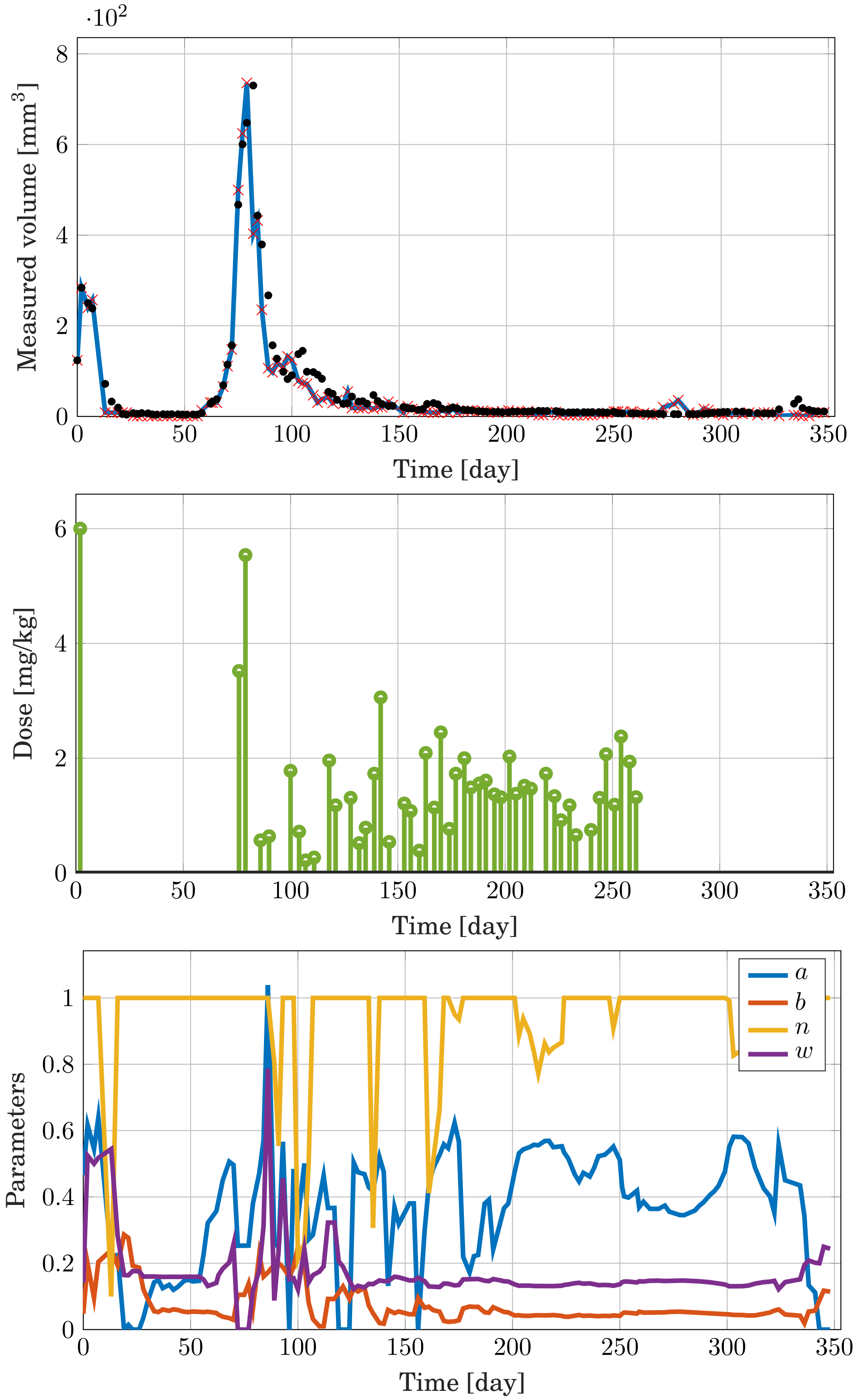


Figure B.12: Evolution of the tumor volume, the corresponding doses and the identified time-varying parameters for $S_{2,12}$. The red crosses represent the tumor volume measurements, while the black dots show their estimations, produced by the MHE.

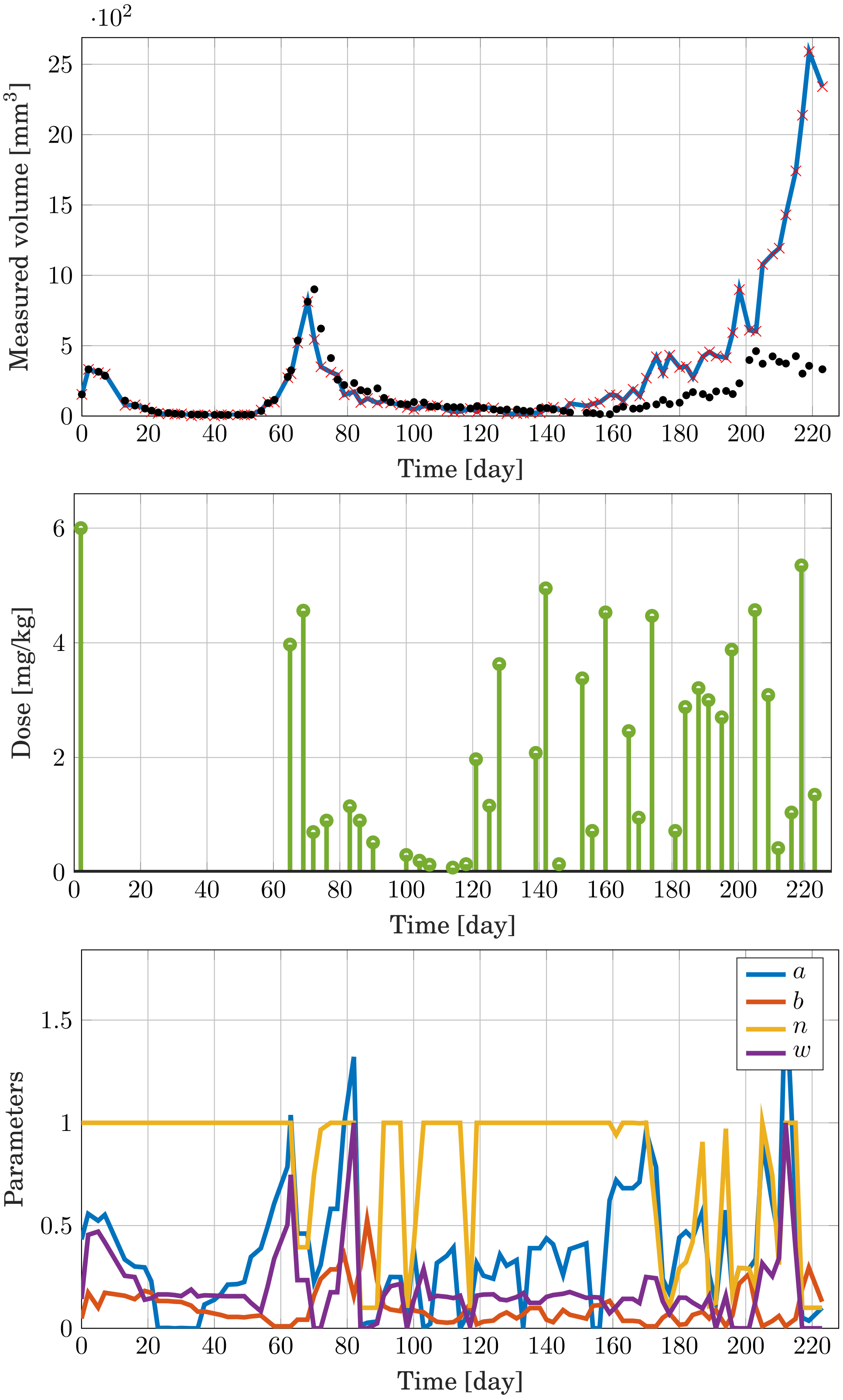


Figure B.13: Evolution of the tumor volume, the corresponding doses and the identified time-varying parameters for $S_{2,13}$. The red crosses represent the tumor volume measurements, while the black dots show their estimations, produced by the MHE.

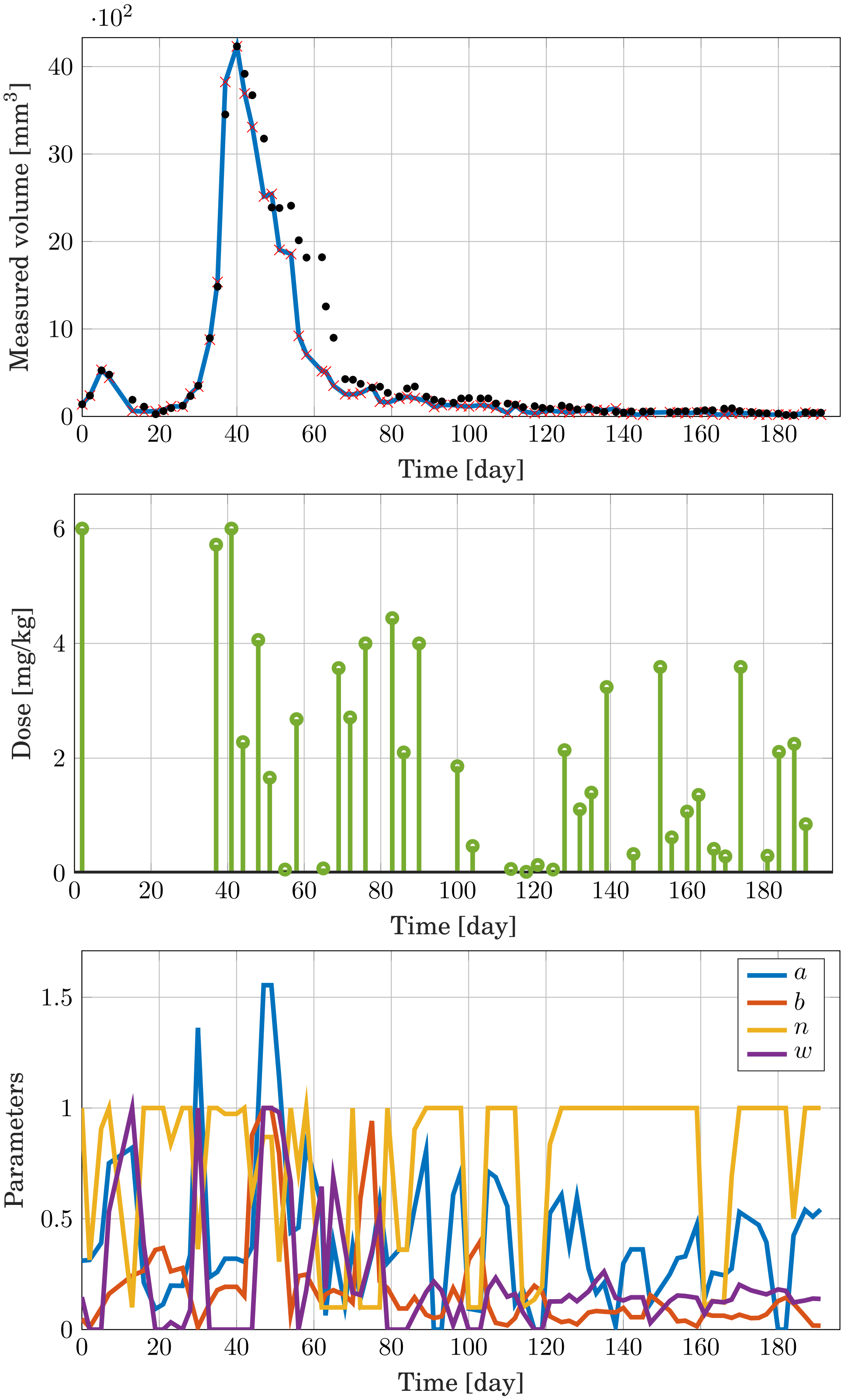


Figure B.14: Evolution of the tumor volume, the corresponding doses and the identified time-varying parameters for $S_{2,14}$. The red crosses represent the tumor volume measurements, while the black dots show their estimations, produced by the MHE.

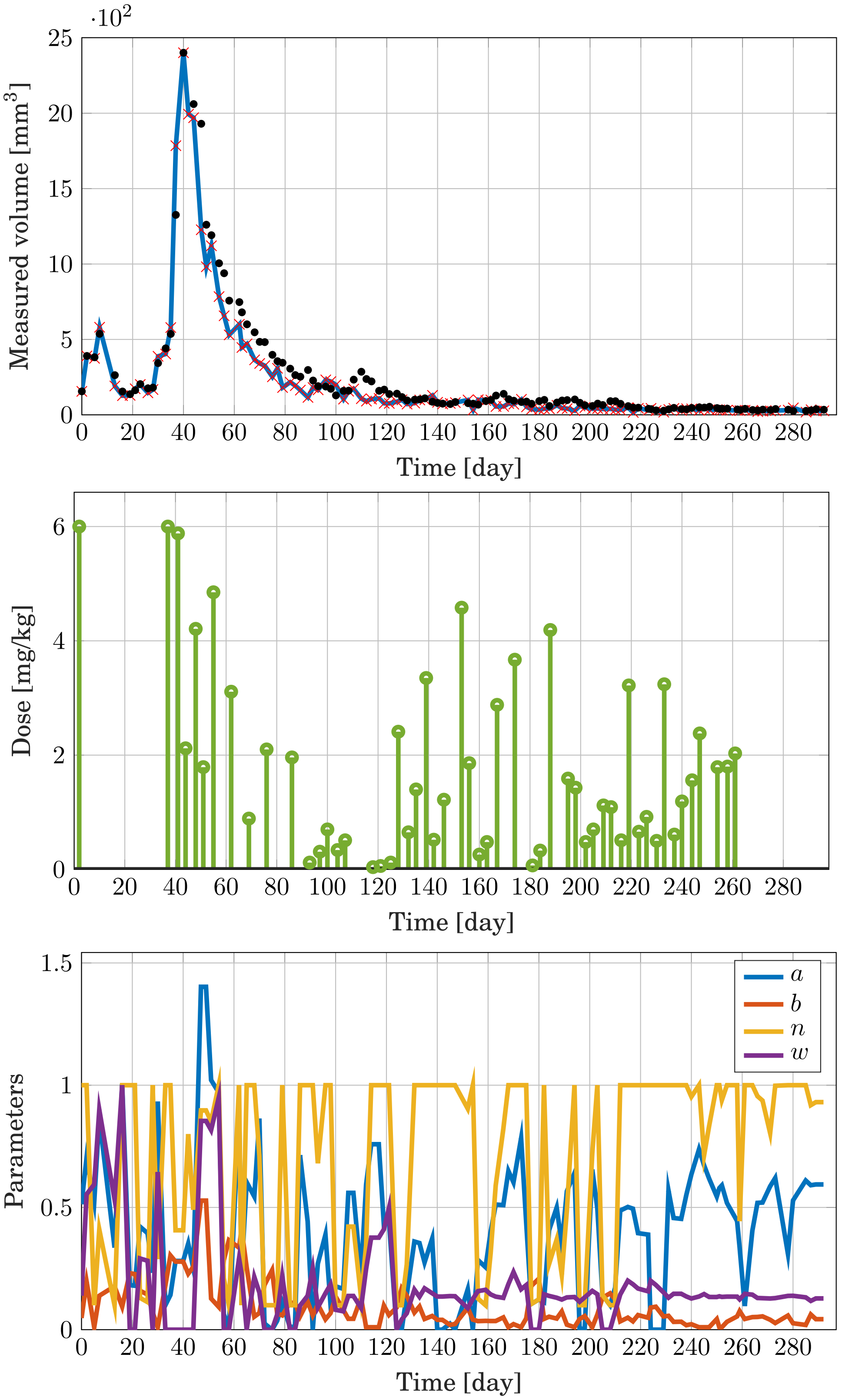


Figure B.15: Evolution of the tumor volume, the corresponding doses and the identified time-varying parameters for $S_{2,15}$. The red crosses represent the tumor volume measurements, while the black dots show their estimations, produced by the MHE.

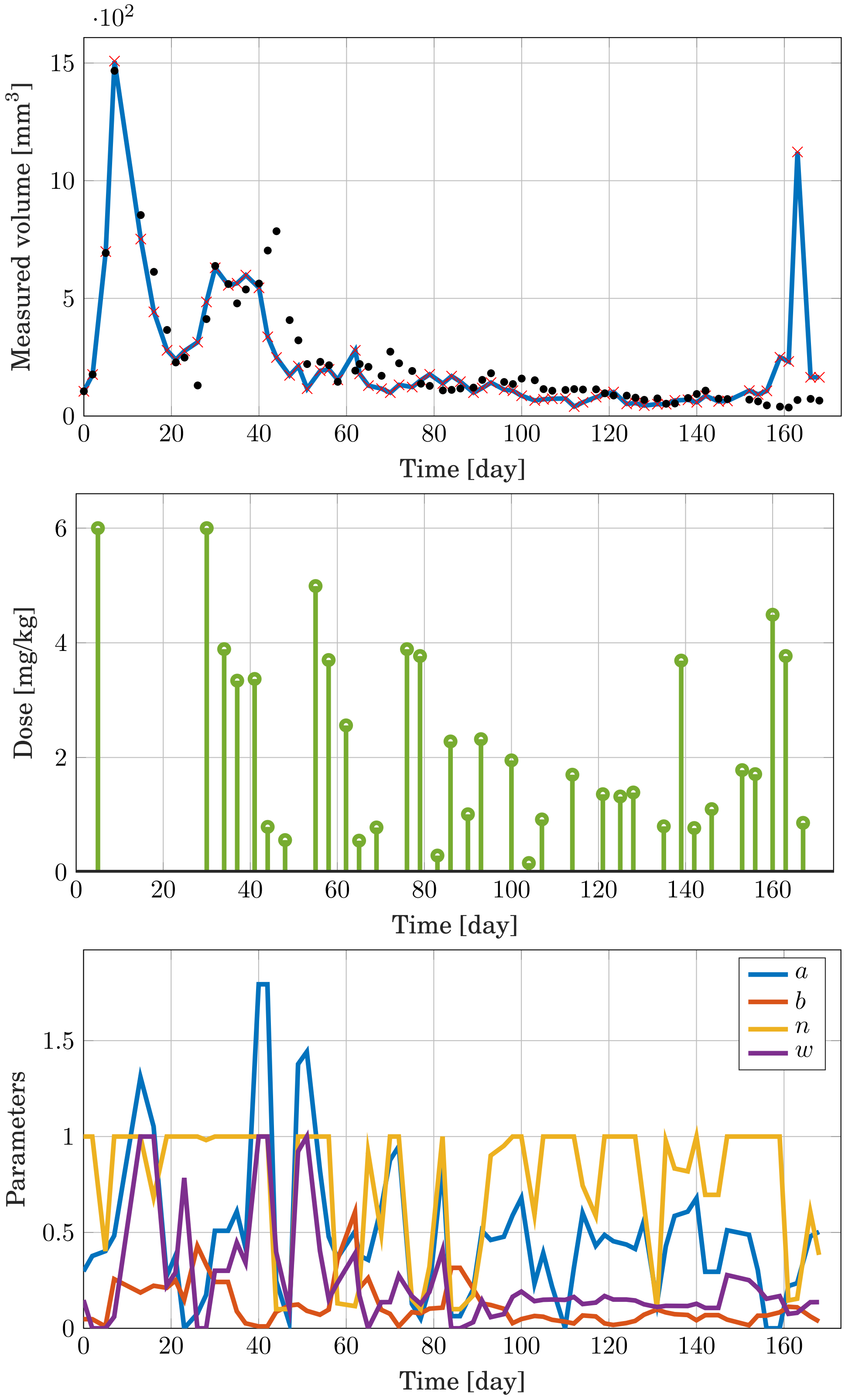


Figure B.16: Evolution of the tumor volume, the corresponding doses and the identified time-varying parameters for $S_{2,16}$. The red crosses represent the tumor volume measurements, while the black dots show their estimations, produced by the MHE.

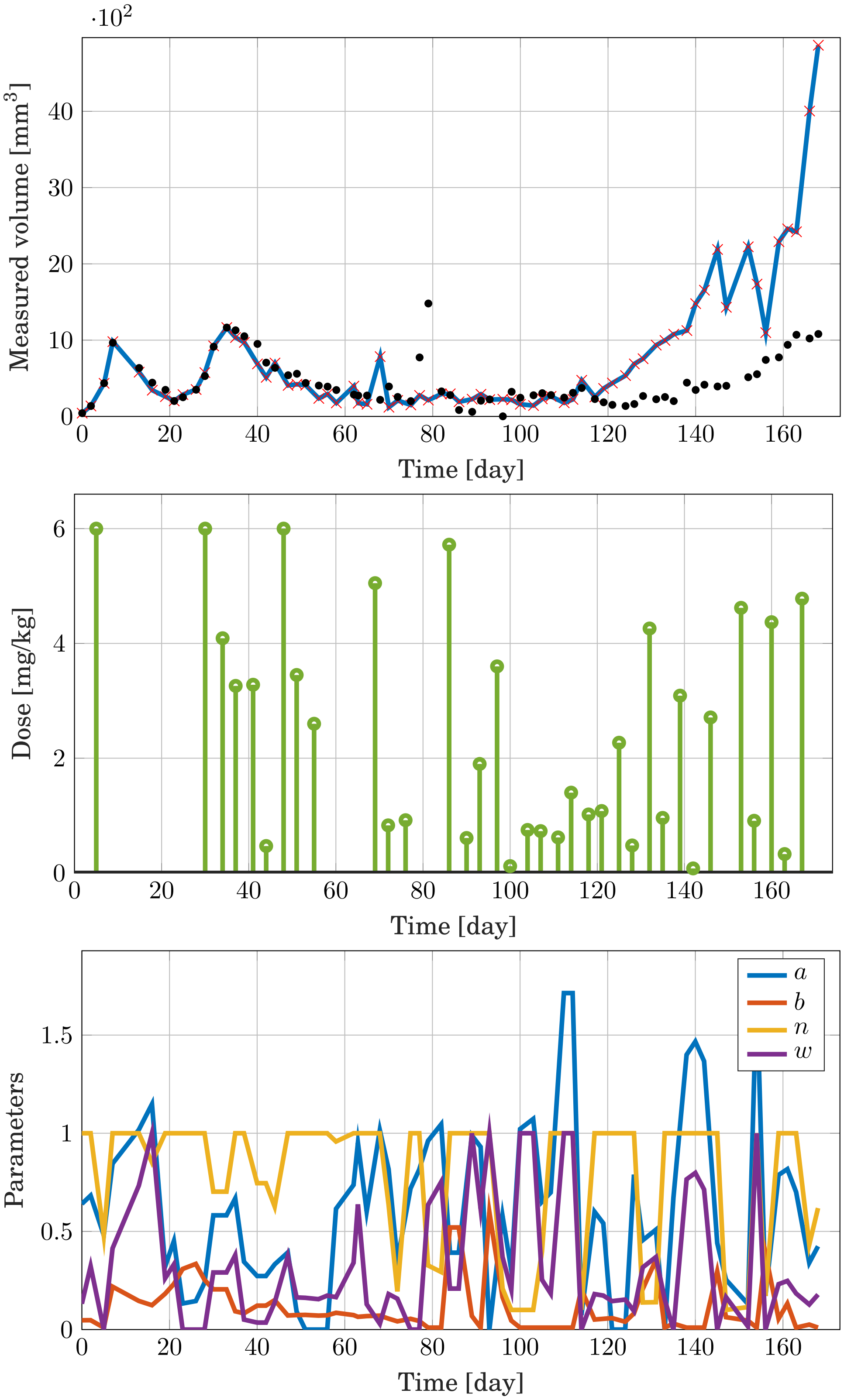


Figure B.17: Evolution of the tumor volume, the corresponding doses and the identified time-varying parameters for $S_{2,17}$. The red crosses represent the tumor volume measurements, while the black dots show their estimations, produced by the MHE.

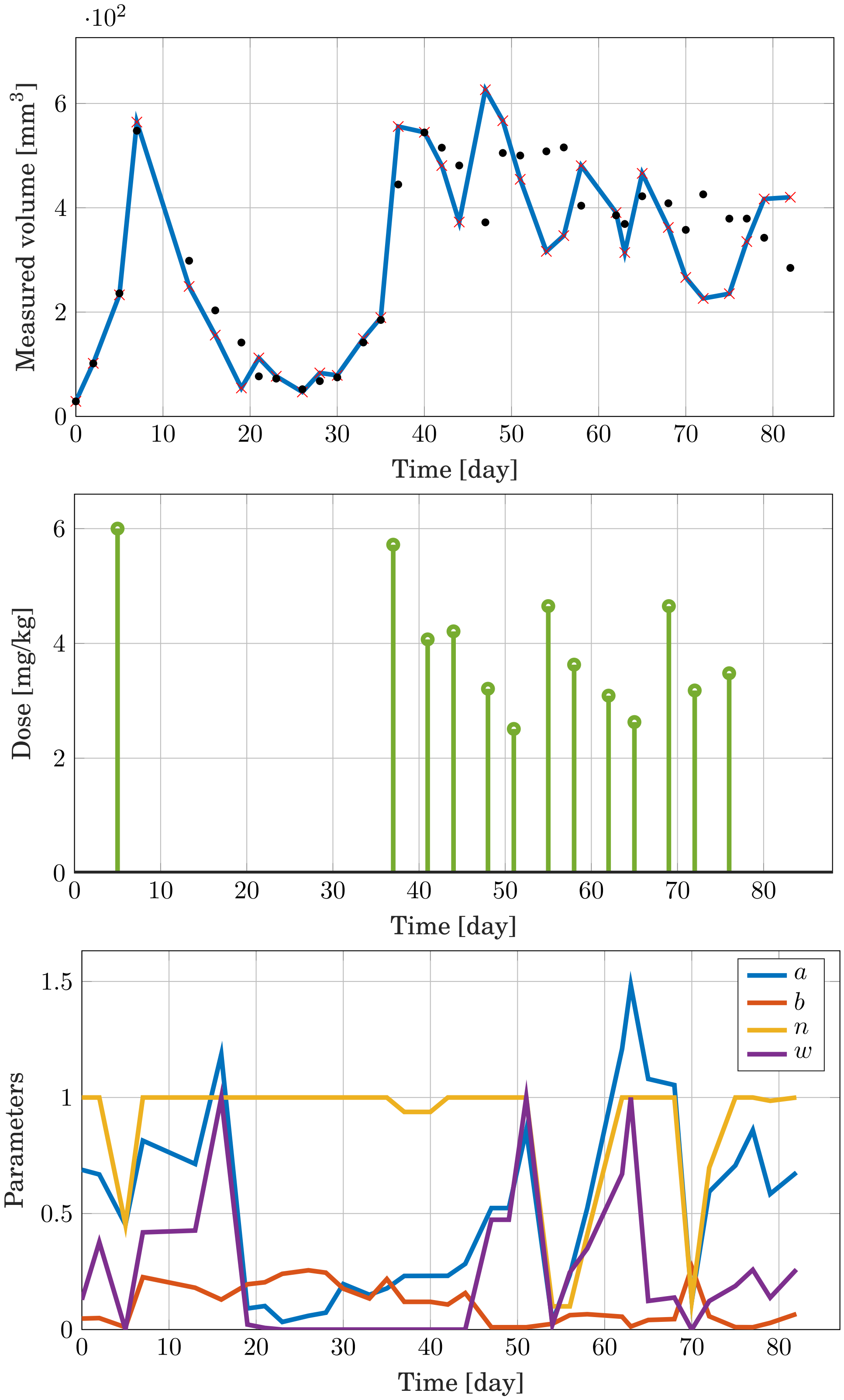


Figure B.18: Evolution of the tumor volume, the corresponding doses and the identified time-varying parameters for $S_{2,18}$. The red crosses represent the tumor volume measurements, while the black dots show their estimations, produced by the MHE.

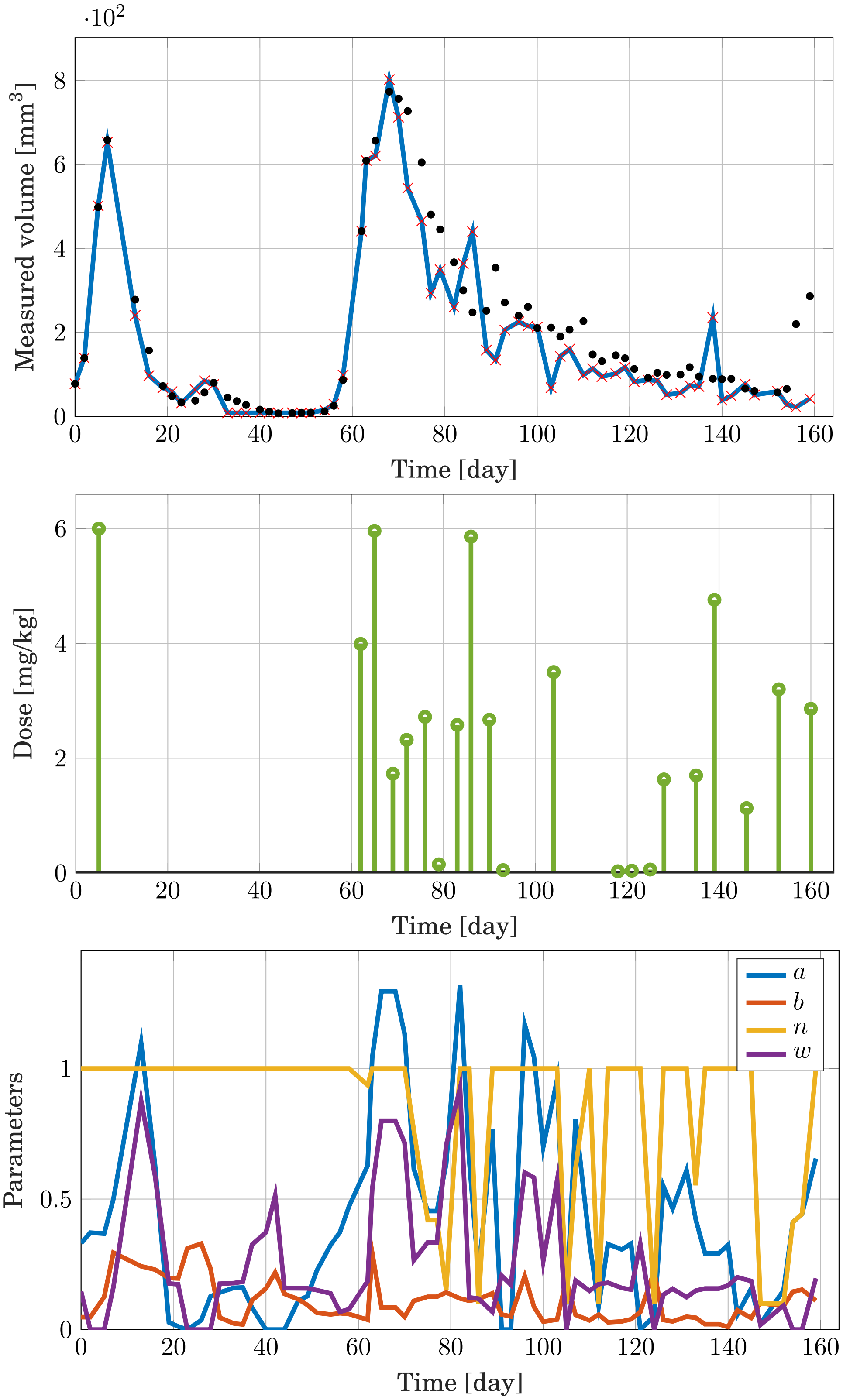


Figure B.19: Evolution of the tumor volume, the corresponding doses and the identified time-varying parameters for $S_{2,19}$. The red crosses represent the tumor volume measurements, while the black dots show their estimations, produced by the MHE.

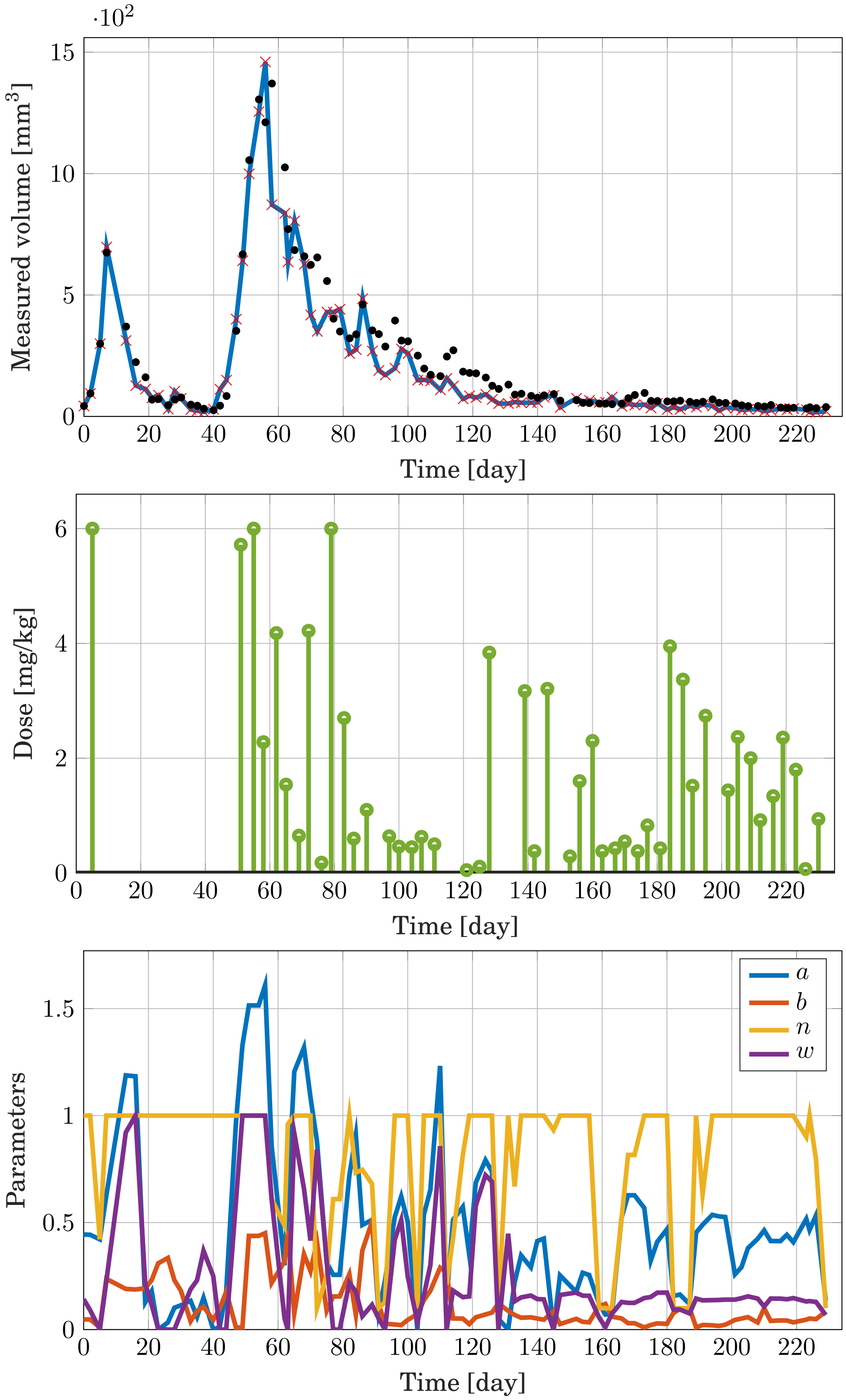


Figure B.20: Evolution of the tumor volume, the corresponding doses and the identified time-varying parameters for $S_{2,20}$. The red crosses represent the tumor volume measurements, while the black dots show their estimations, produced by the MHE.

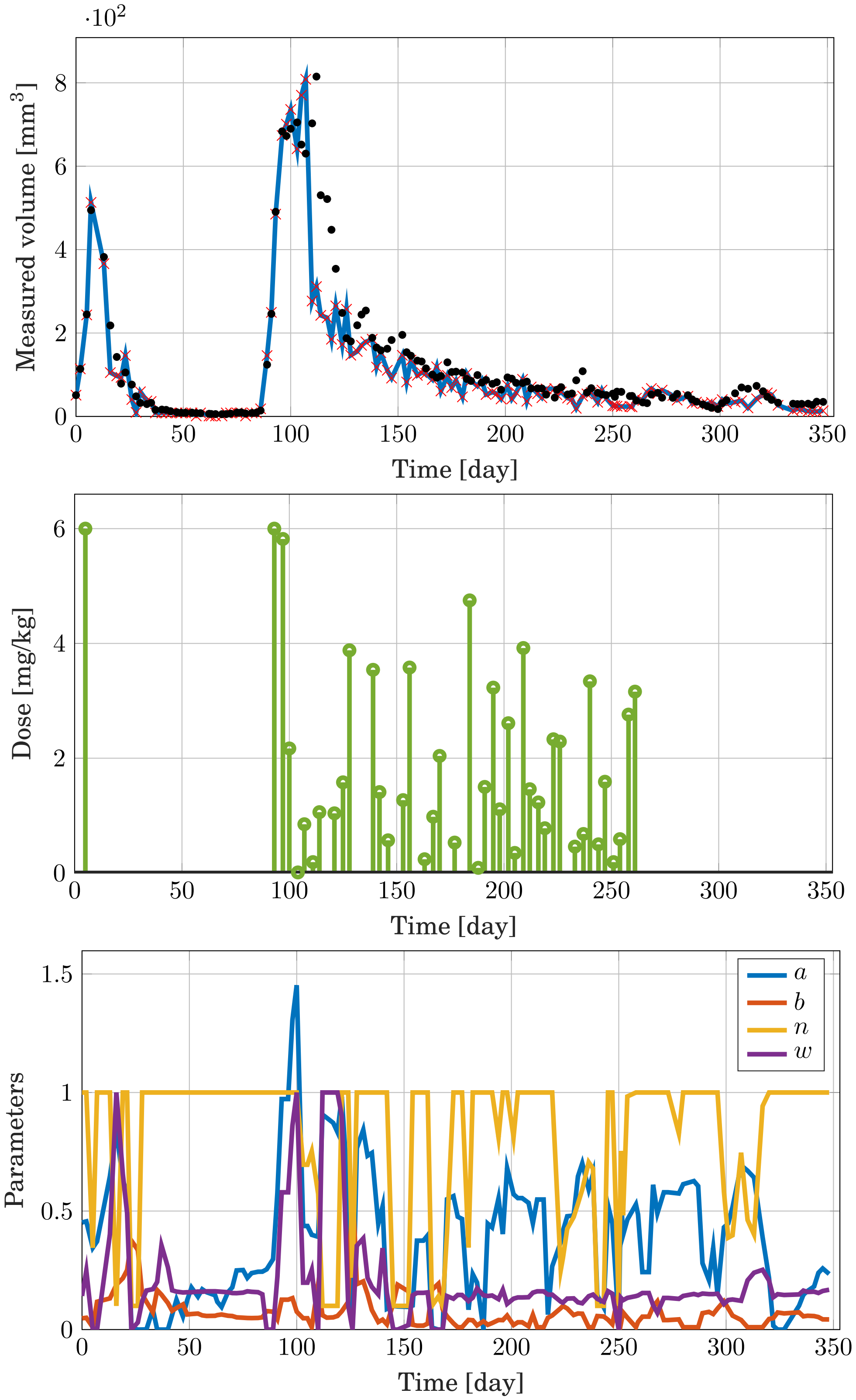


Figure B.21: Evolution of the tumor volume, the corresponding doses and the identified time-varying parameters for $S_{2,21}$. The red crosses represent the tumor volume measurements, while the black dots show their estimations, produced by the MHE.

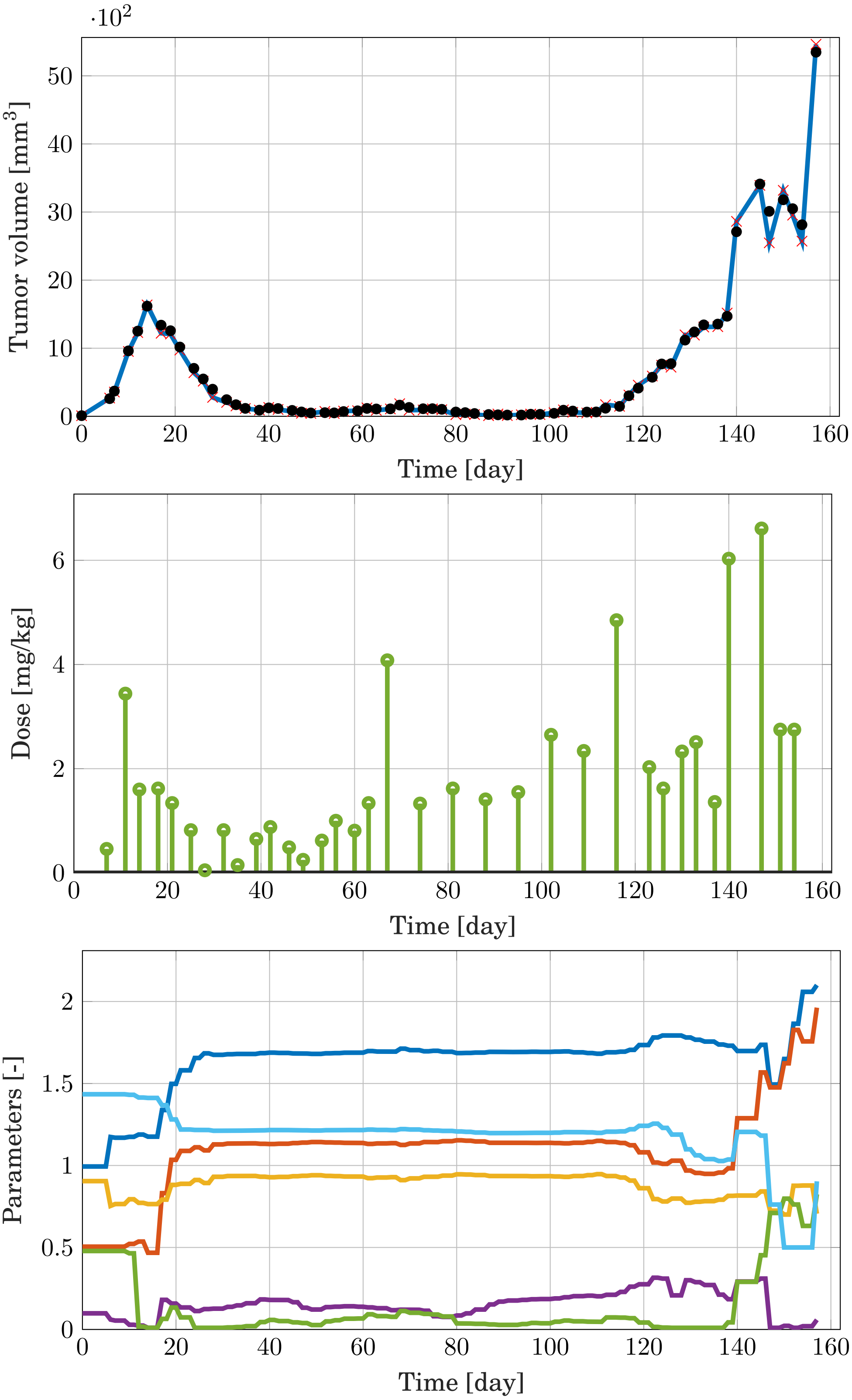


Figure B.22: Evolution of the tumor volume, the corresponding doses and the identified time-varying parameters for $S_{3,1}$. The red crosses represent the tumor volume measurements, while the black dots show their estimations, produced by the MHE. For the parameters a, b, n, w, ED_{50} and c in the bottom figure, the colors blue, red, yellow, purple, green, and turquoise are assigned, respectively.

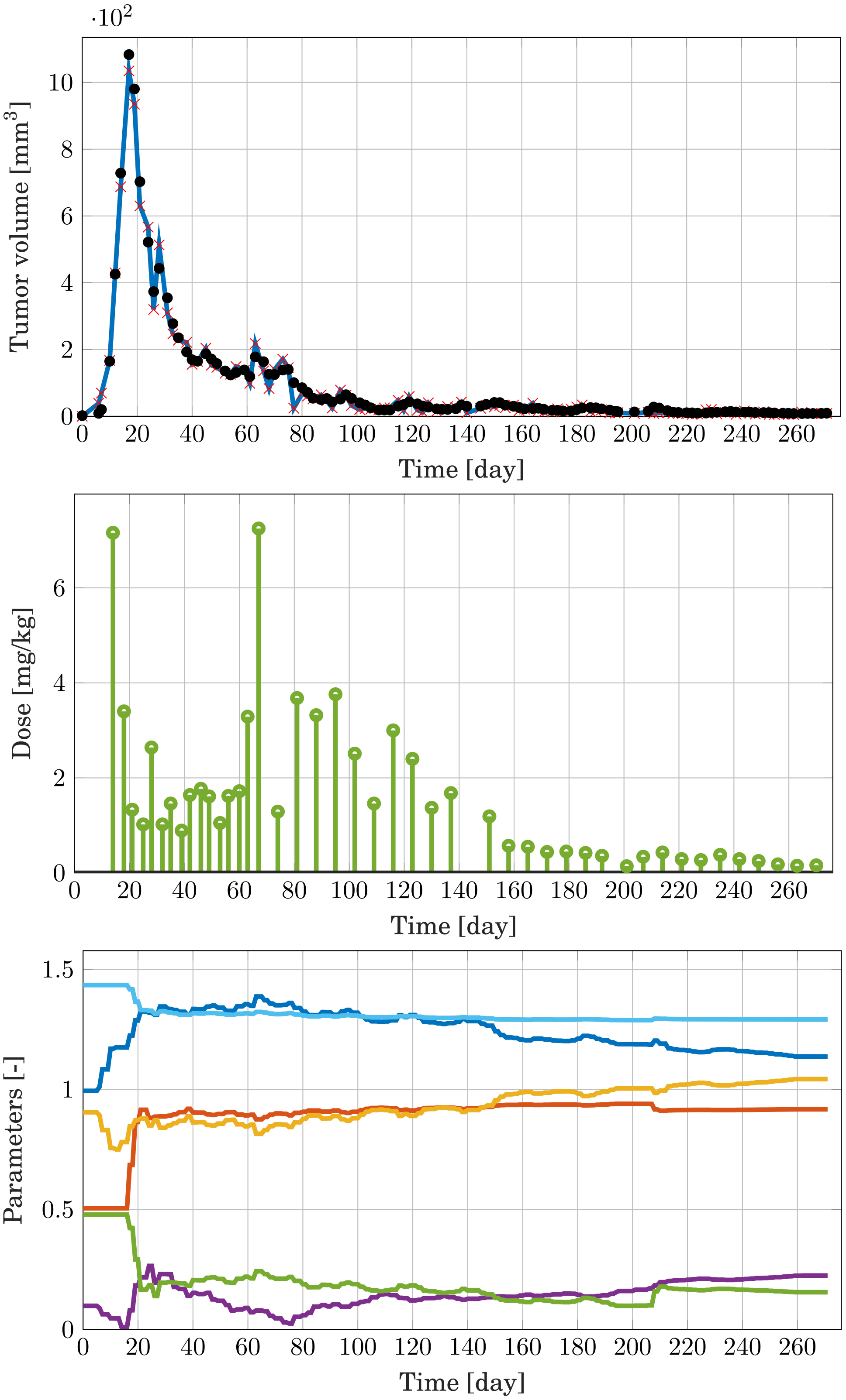


Figure B.23: Evolution of the tumor volume, the corresponding doses and the identified time-varying parameters for $S_{3,2}$. The red crosses represent the tumor volume measurements, while the black dots show their estimations, produced by the MHE. For the parameters a, b, n, w, ED_{50} and c in the bottom figure, the colors blue, red, yellow, purple, green, and turquoise are assigned, respectively.

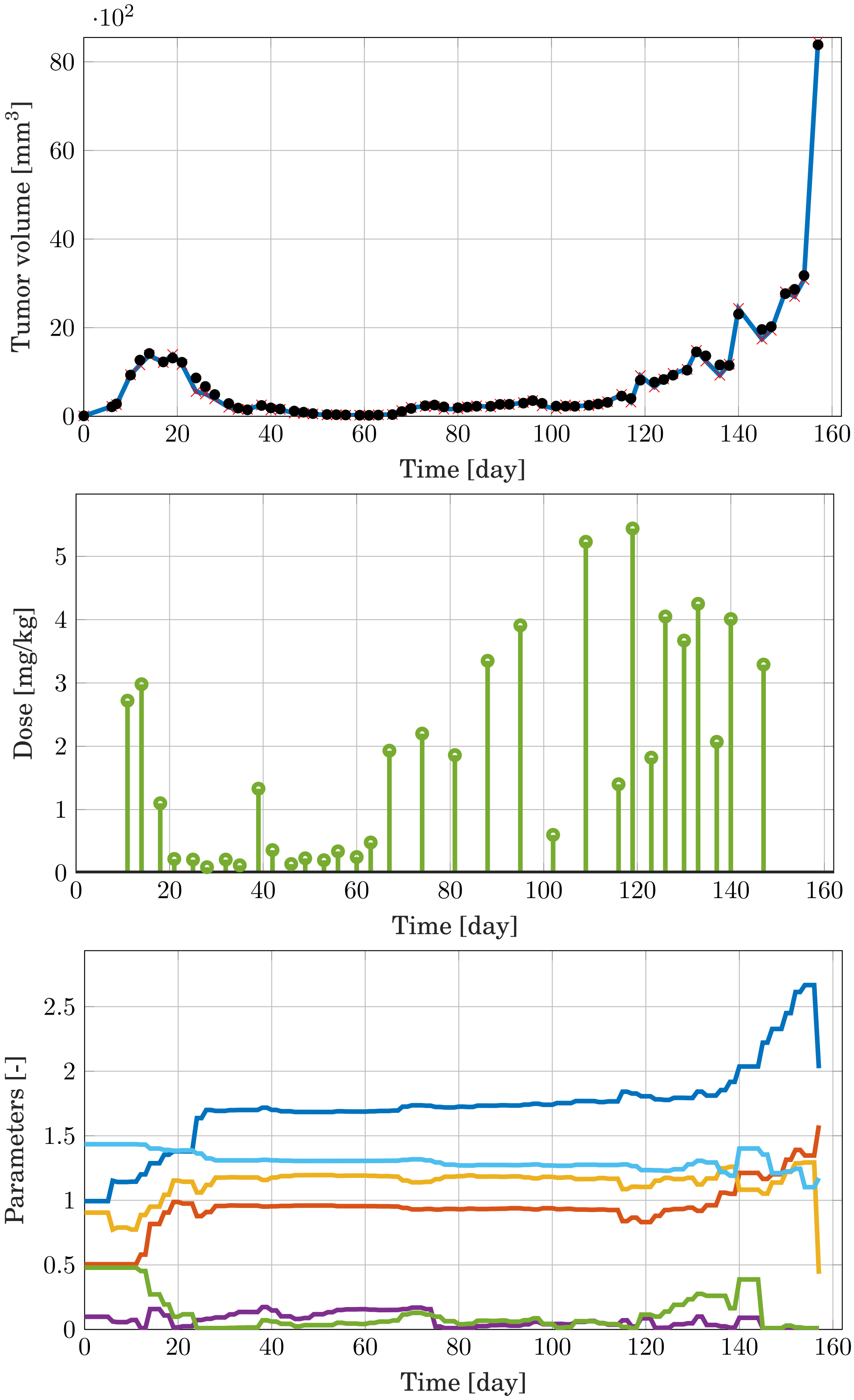


Figure B.24: Evolution of the tumor volume, the corresponding doses and the identified time-varying parameters for $S_{3,3}$. The red crosses represent the tumor volume measurements, while the black dots show their estimations, produced by the MHE. For the parameters a, b, n, w, ED_{50} and c in the bottom figure, the colors blue, red, yellow, purple, green, and turquoise are assigned, respectively.

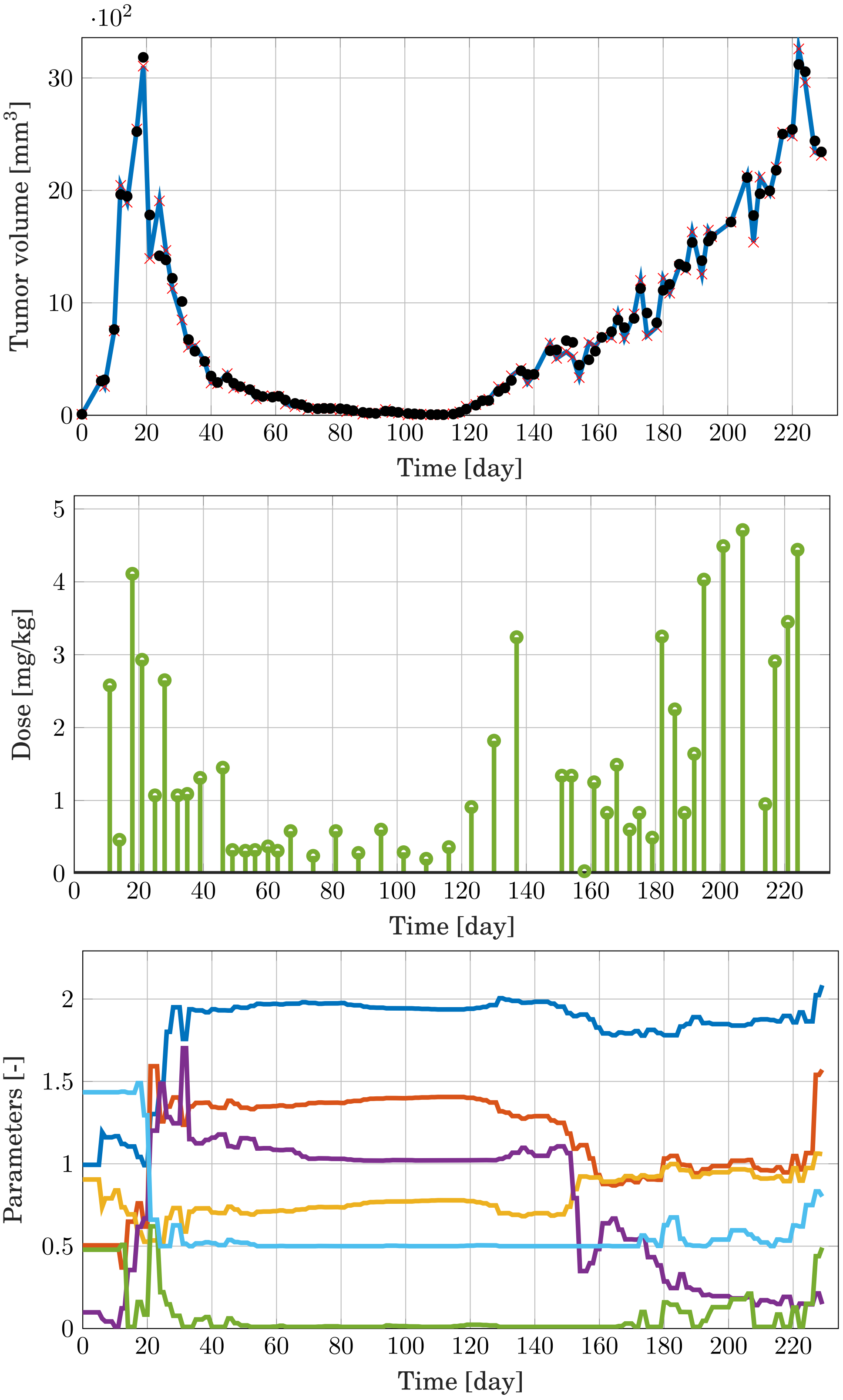


Figure B.25: Evolution of the tumor volume, the corresponding doses and the identified time-varying parameters for $S_{3,4}$. The red crosses represent the tumor volume measurements, while the black dots show their estimations, produced by the MHE. For the parameters a, b, n, w, ED_{50} and c in the bottom figure, the colors blue, red, yellow, purple, green, and turquoise are assigned, respectively.

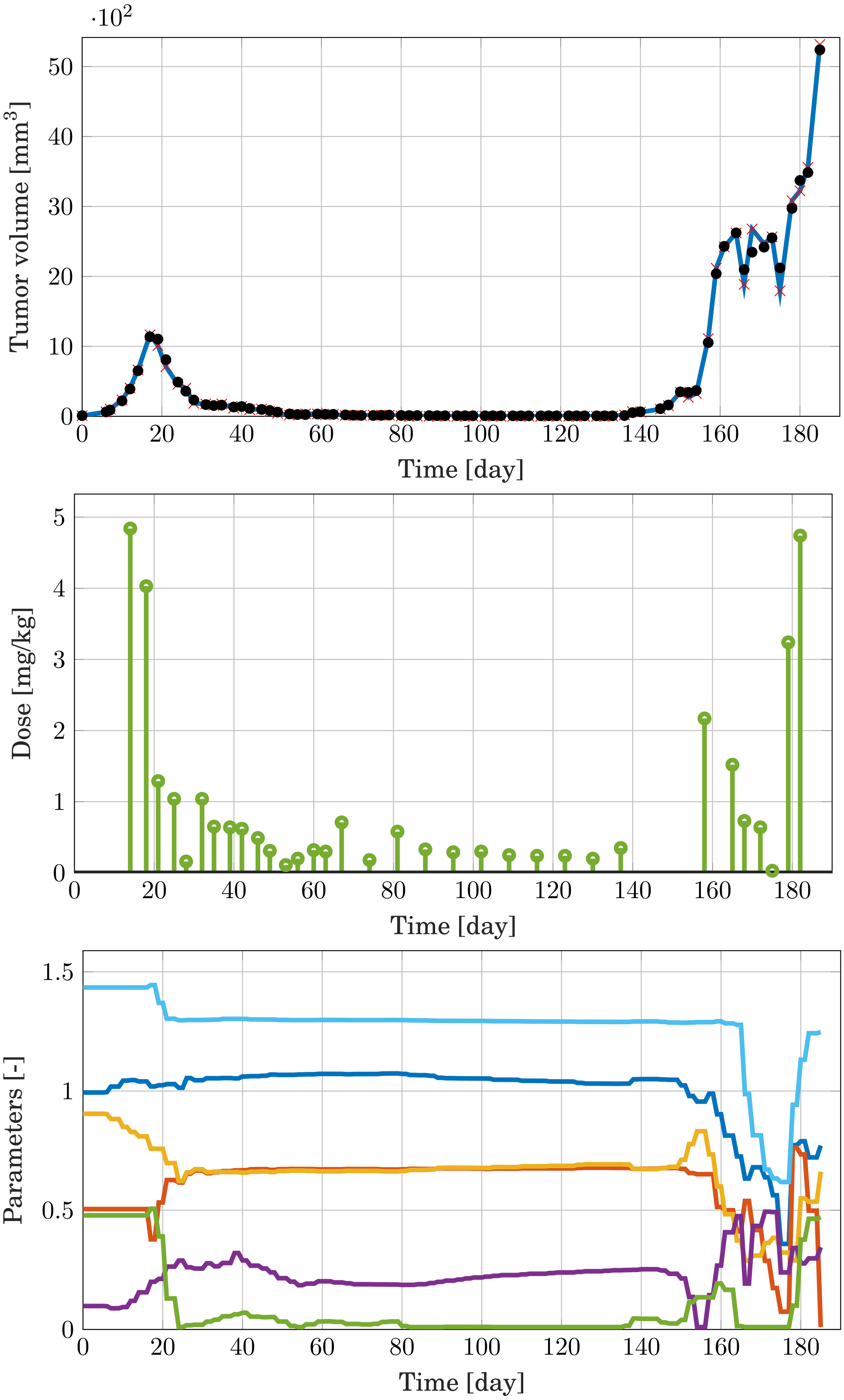


Figure B.26: Evolution of the tumor volume, the corresponding doses and the identified time-varying parameters for $S_{3,6}$. The red crosses represent the tumor volume measurements, while the black dots show their estimations, produced by the MHE. For the parameters a, b, n, w, ED_{50} and c in the bottom figure, the colors blue, red, yellow, purple, green, and turquoise are assigned, respectively.

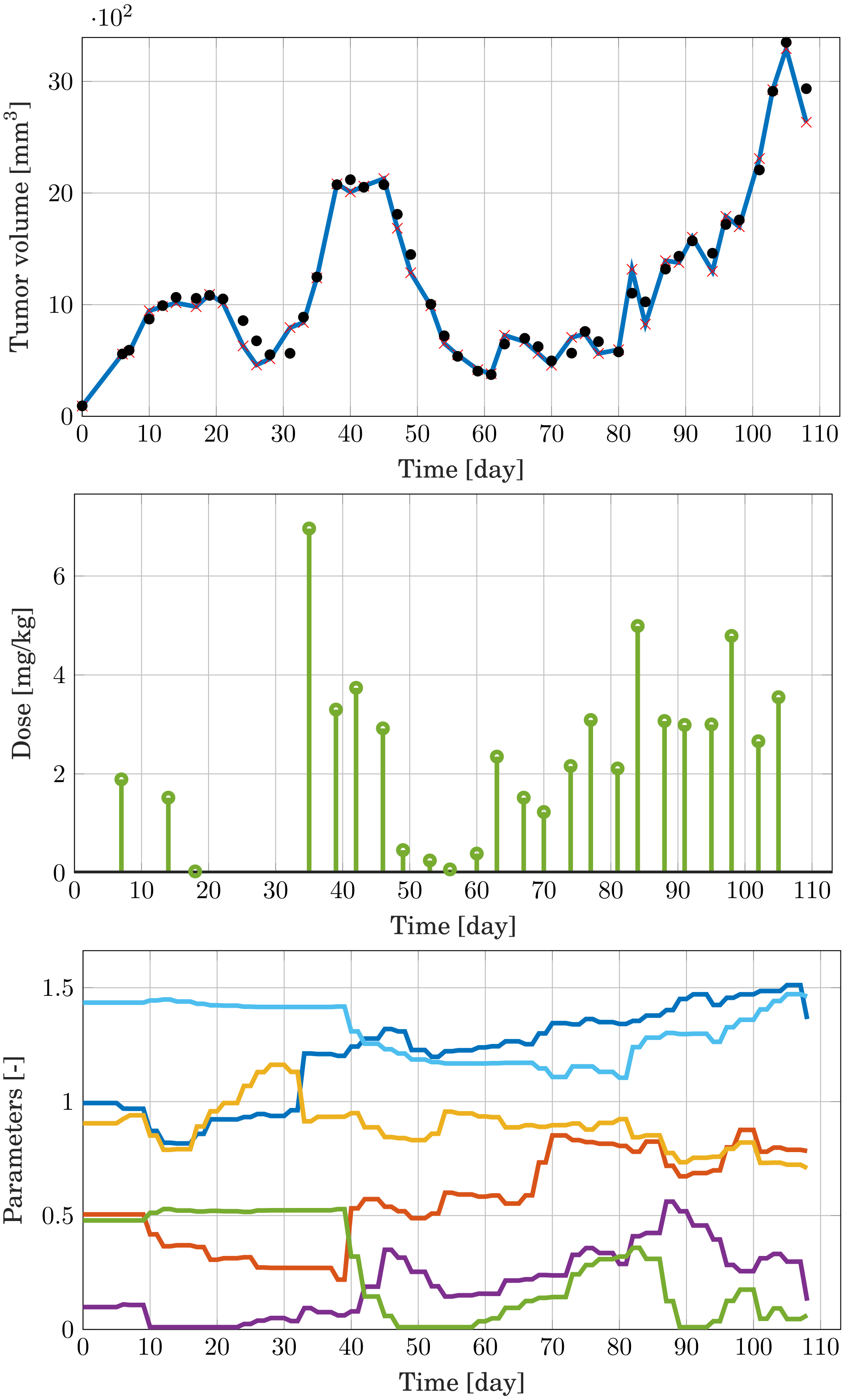


Figure B.27: Evolution of the tumor volume, the corresponding doses and the identified time-varying parameters for $S_{3,7}$. The red crosses represent the tumor volume measurements, while the black dots show their estimations, produced by the MHE. For the parameters a, b, n, w, ED_{50} and c in the bottom figure, the colors blue, red, yellow, purple, green, and turquoise are assigned, respectively.

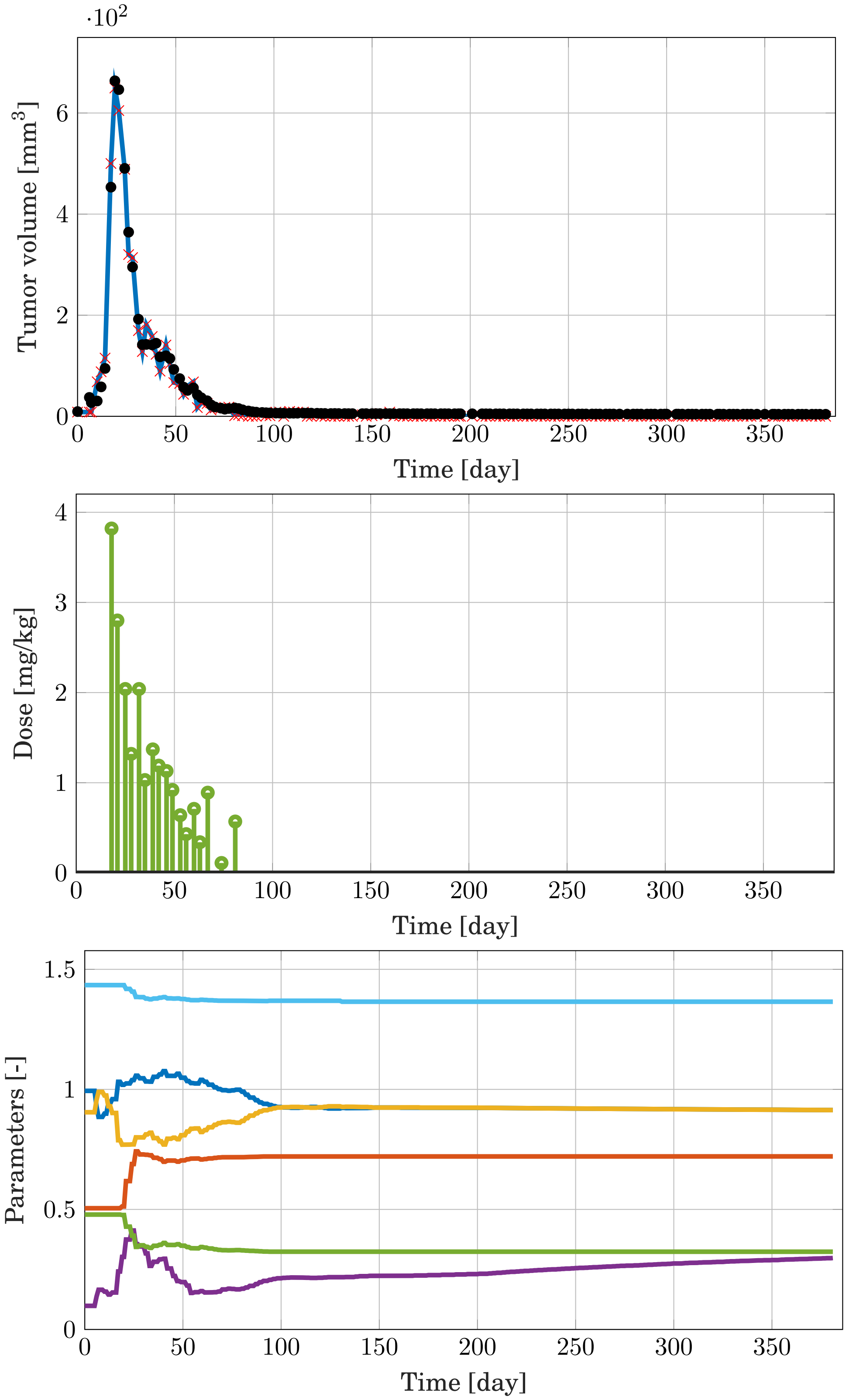


Figure B.28: Evolution of the tumor volume, the corresponding doses and the identified time-varying parameters for $S_{3,8}$. The red crosses represent the tumor volume measurements, while the black dots show their estimations, produced by the MHE. For the parameters a, b, n, w, ED_{50} and c in the bottom figure, the colors blue, red, yellow, purple, green, and turquoise are assigned, respectively.

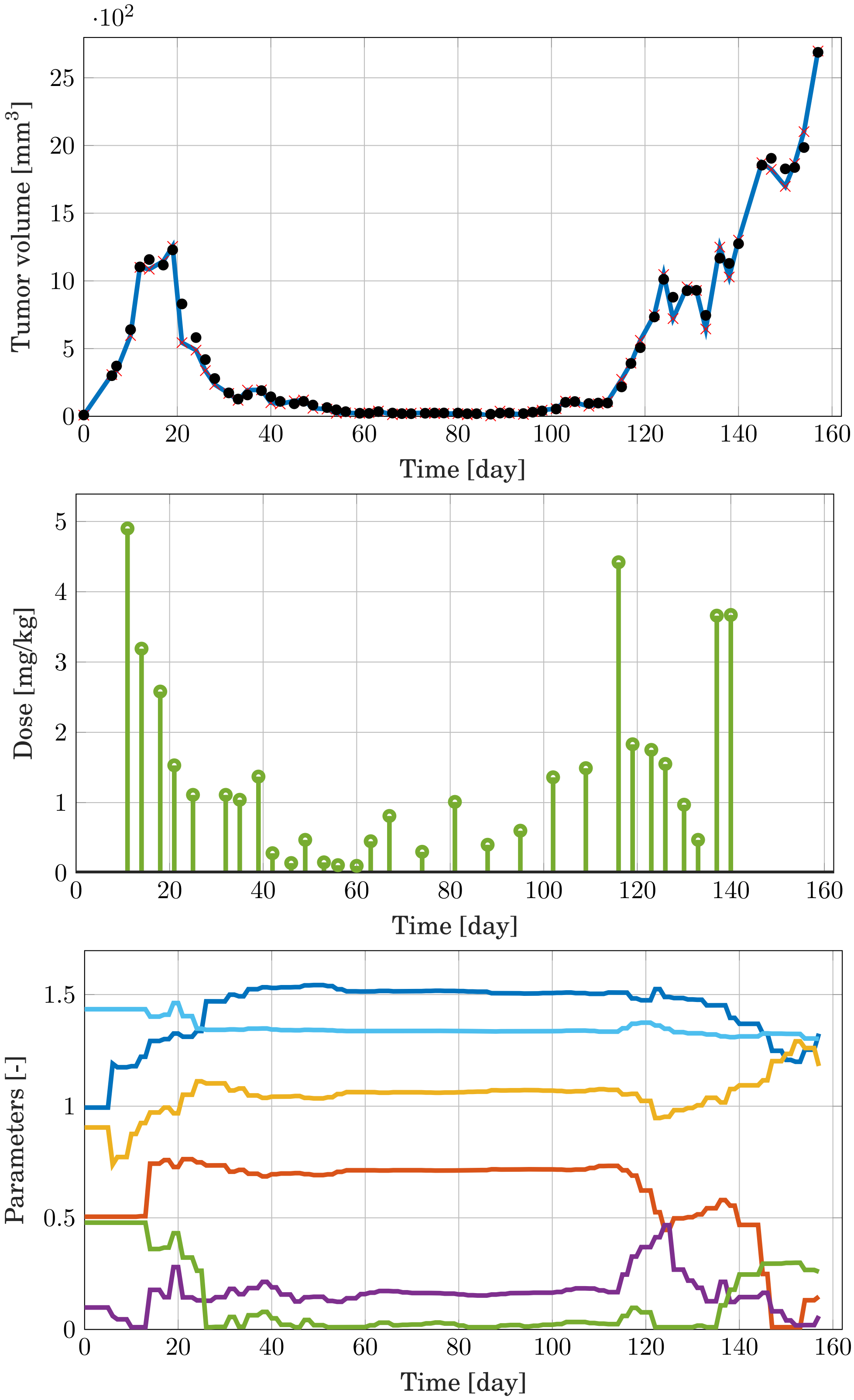


Figure B.29: Evolution of the tumor volume, the corresponding doses and the identified time-varying parameters for $S_{3,9}$. The red crosses represent the tumor volume measurements, while the black dots show their estimations, produced by the MHE. For the parameters a, b, n, w, ED_{50} and c in the bottom figure, the colors blue, red, yellow, purple, green, and turquoise are assigned, respectively.

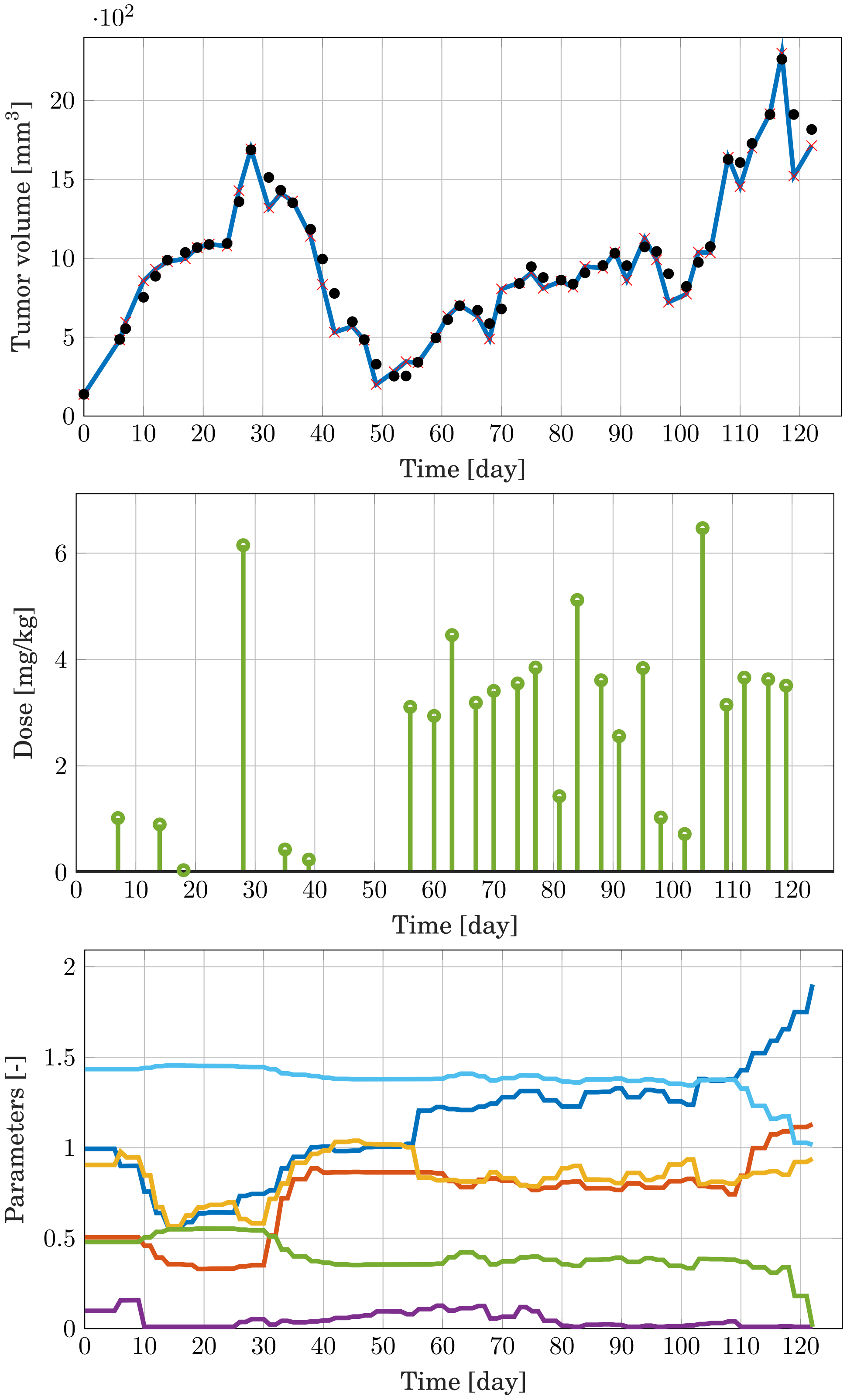


Figure B.30: Evolution of the tumor volume, the corresponding doses and the identified time-varying parameters for $S_{3,10}$. The red crosses represent the tumor volume measurements, while the black dots show their estimations, produced by the MHE. For the parameters a, b, n, w, ED_{50} and c in the bottom figure, the colors blue, red, yellow, purple, green, and turquoise are assigned, respectively.

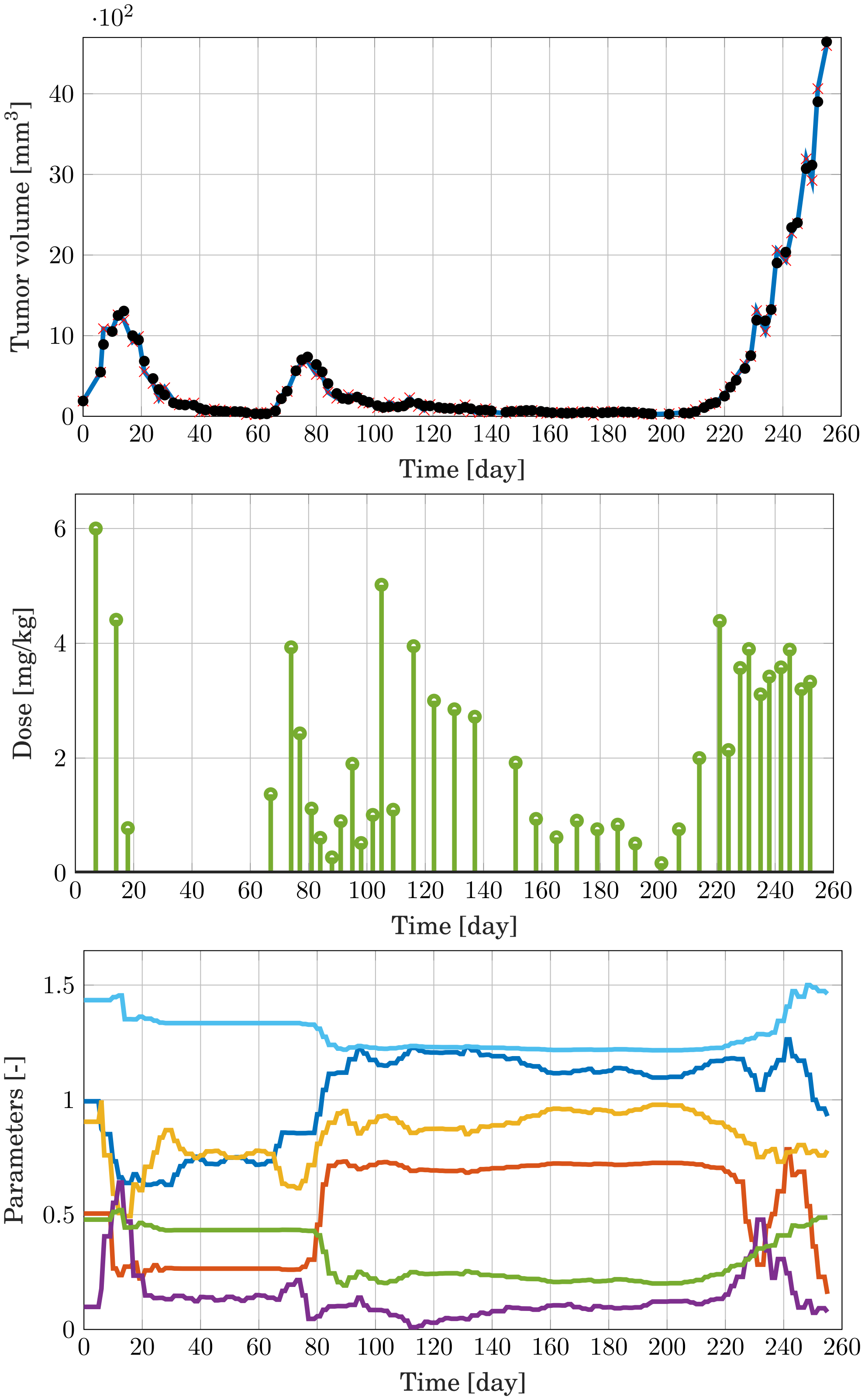


Figure B.31: Evolution of the tumor volume, the corresponding doses and the identified time-varying parameters for $S_{3,11}$. The red crosses represent the tumor volume measurements, while the black dots show their estimations, produced by the MHE. For the parameters a, b, n, w, ED_{50} and c in the bottom figure, the colors blue, red, yellow, purple, green, and turquoise are assigned, respectively.

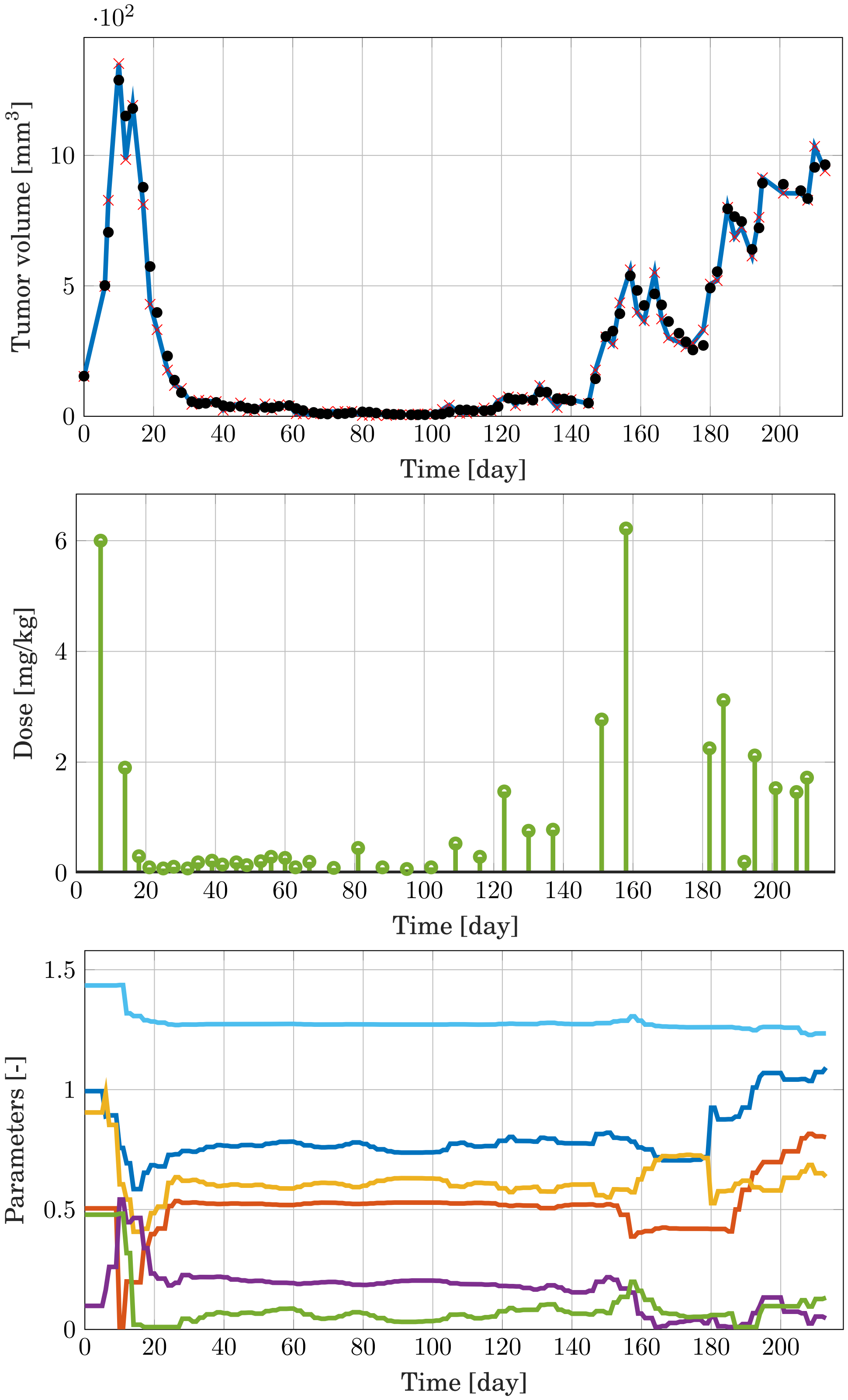


Figure B.32: Evolution of the tumor volume, the corresponding doses and the identified time-varying parameters for $S_{3,12}$. The red crosses represent the tumor volume measurements, while the black dots show their estimations, produced by the MHE. For the parameters a, b, n, w, ED_{50} and c in the bottom figure, the colors blue, red, yellow, purple, green, and turquoise are assigned, respectively.

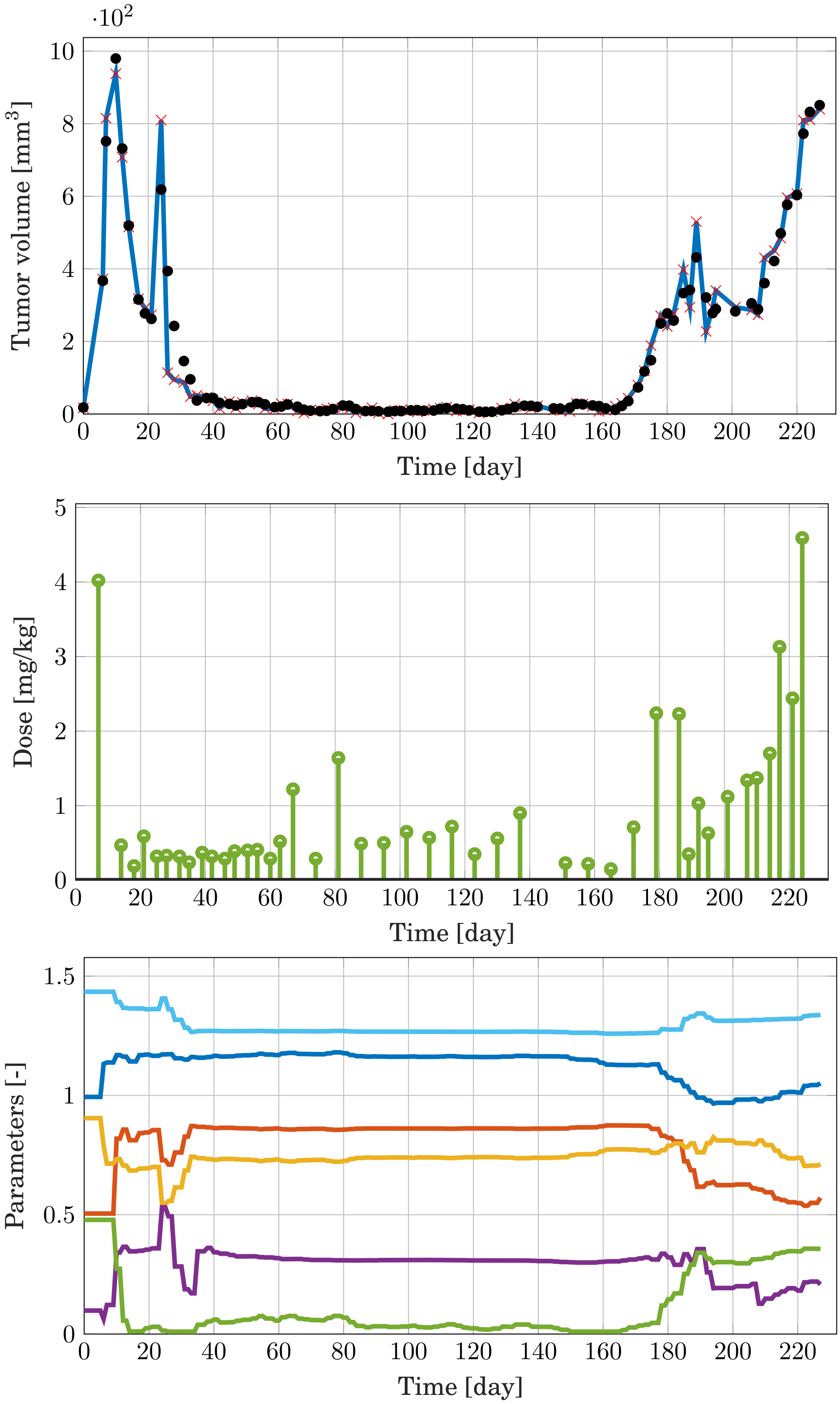


Figure B.33: Evolution of the tumor volume, the corresponding doses and the identified time-varying parameters for $S_{3,13}$. The red crosses represent the tumor volume measurements, while the black dots show their estimations, produced by the MHE. For the parameters a, b, n, w, ED_{50} and c in the bottom figure, the colors blue, red, yellow, purple, green, and turquoise are assigned, respectively.

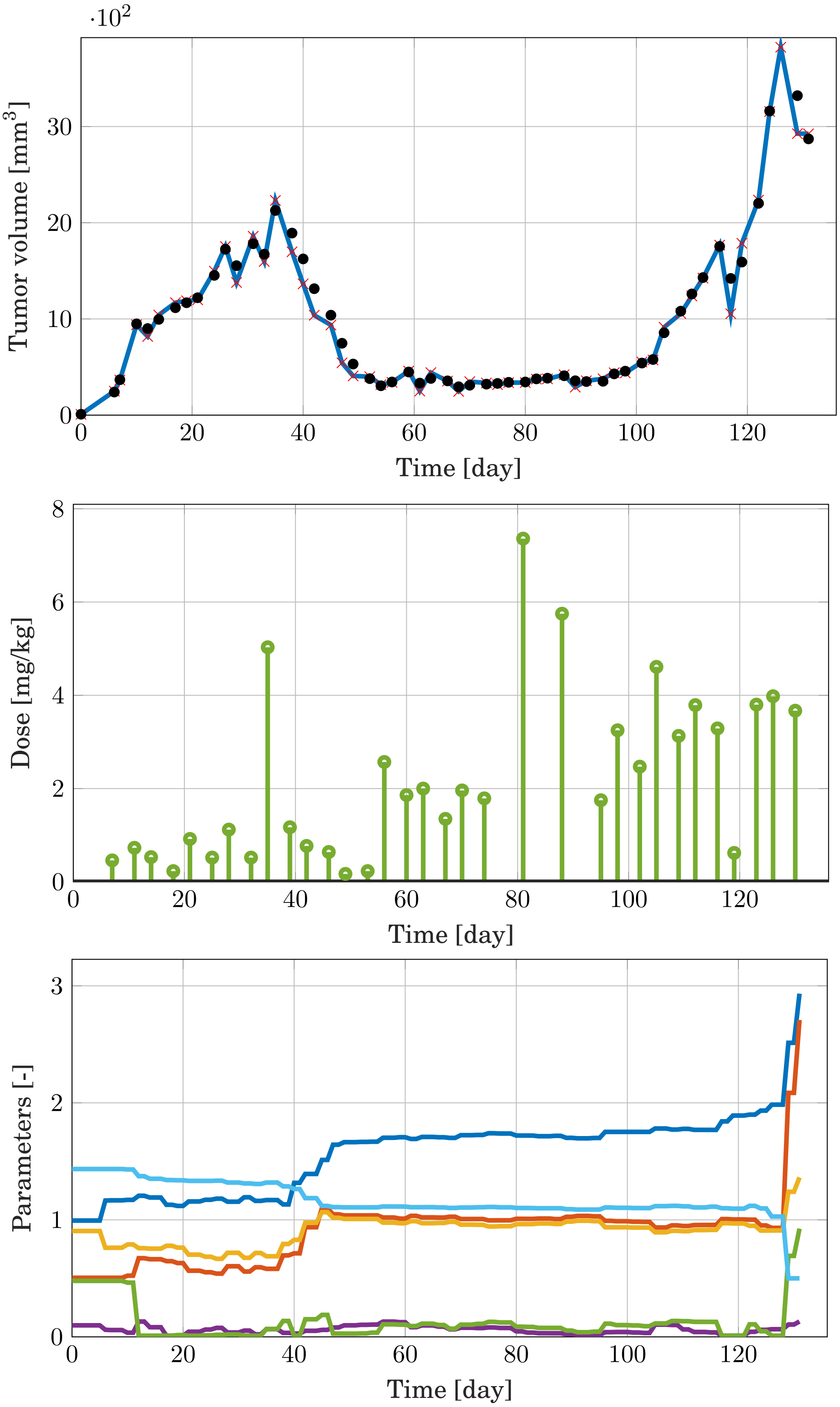


Figure B.34: Evolution of the tumor volume, the corresponding doses and the identified time-varying parameters for $S_{3,14}$. The red crosses represent the tumor volume measurements, while the black dots show their estimations, produced by the MHE. For the parameters a, b, n, w, ED_{50} and c in the bottom figure, the colors blue, red, yellow, purple, green, and turquoise are assigned, respectively.

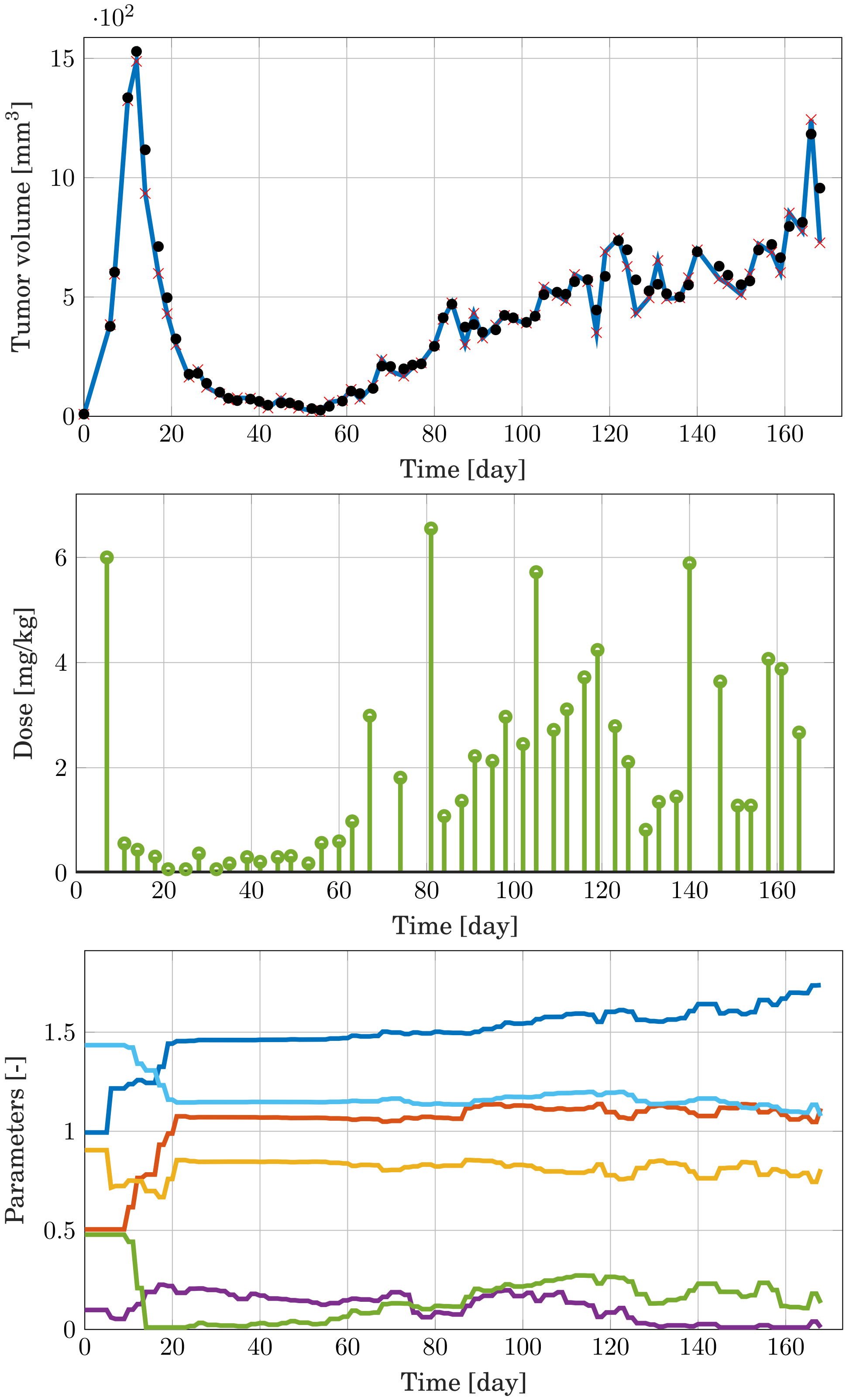


Figure B.35: Evolution of the tumor volume, the corresponding doses and the identified time-varying parameters for $S_{3,15}$. The red crosses represent the tumor volume measurements, while the black dots show their estimations, produced by the MHE. For the parameters a, b, n, w, ED_{50} and c in the bottom figure, the colors blue, red, yellow, purple, green, and turquoise are assigned, respectively.

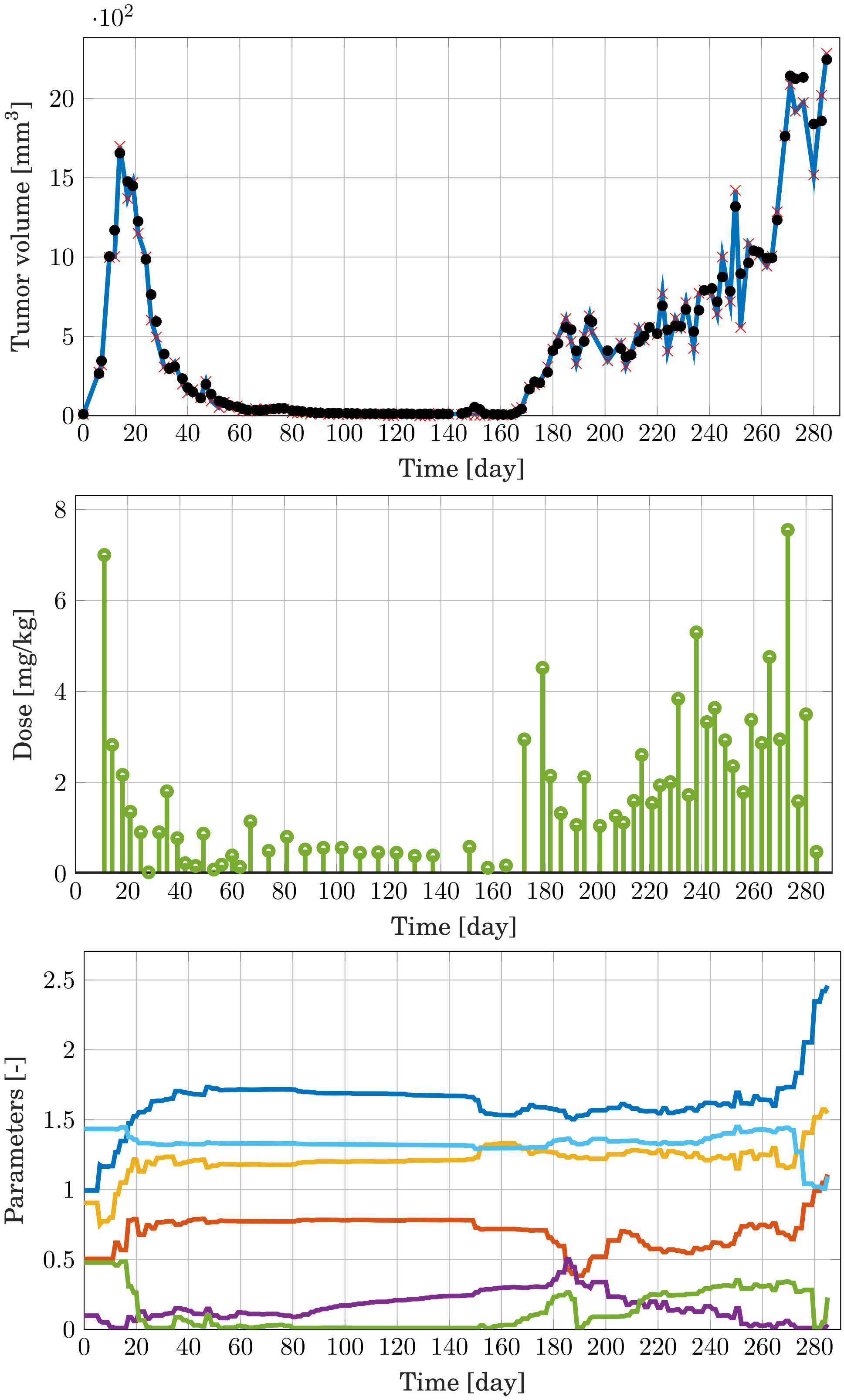


Figure B.36: Evolution of the tumor volume, the corresponding doses and the identified time-varying parameters for $S_{3,16}$. The red crosses represent the tumor volume measurements, while the black dots show their estimations, produced by the MHE. For the parameters a, b, n, w, ED_{50} and c in the bottom figure, the colors blue, red, yellow, purple, green, and turquoise are assigned, respectively.

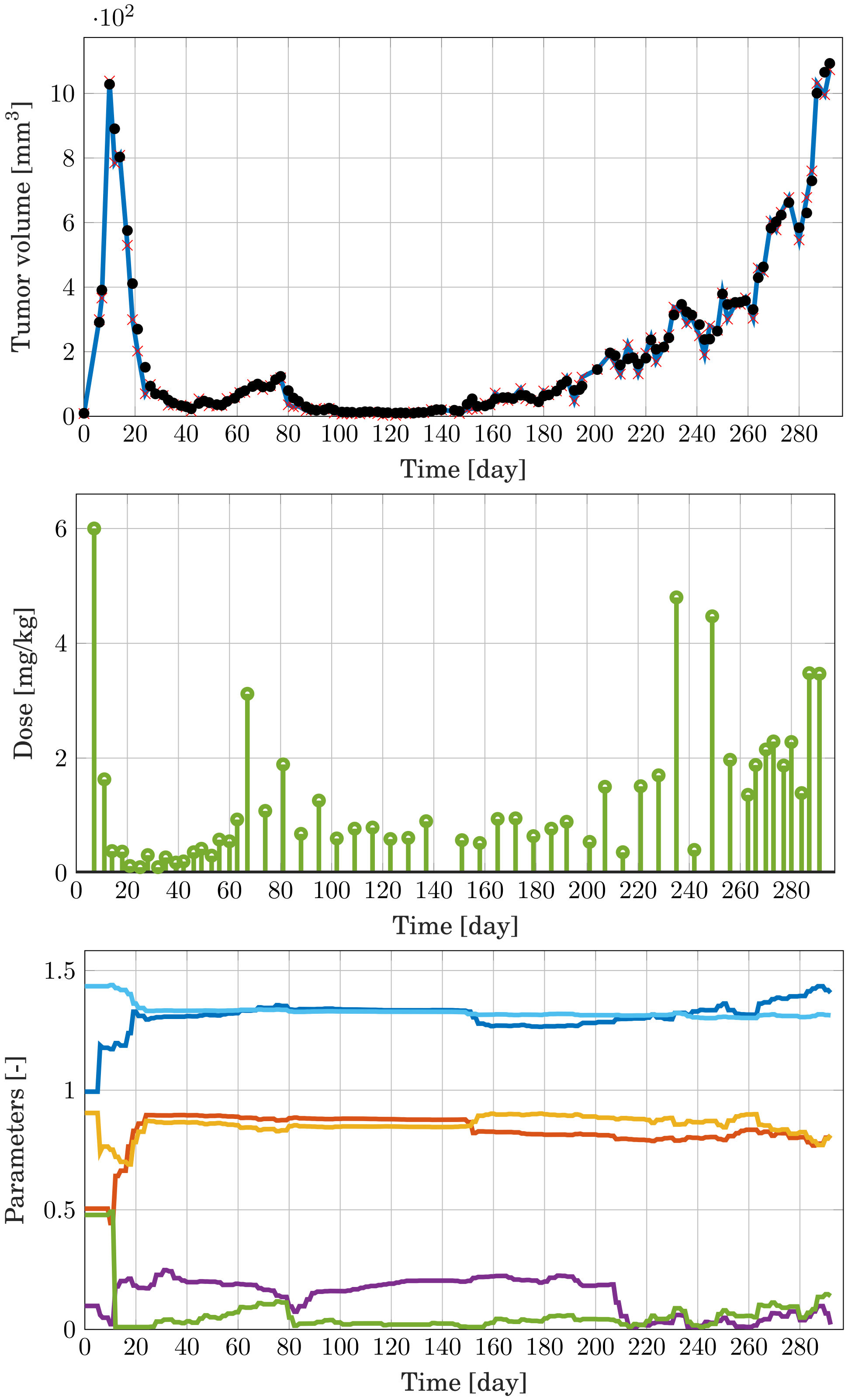


Figure B.37: Evolution of the tumor volume, the corresponding doses and the identified time-varying parameters for $S_{3,17}$. The red crosses represent the tumor volume measurements, while the black dots show their estimations, produced by the MHE. For the parameters a, b, n, w, ED_{50} and c in the bottom figure, the colors blue, red, yellow, purple, green, and turquoise are assigned, respectively.

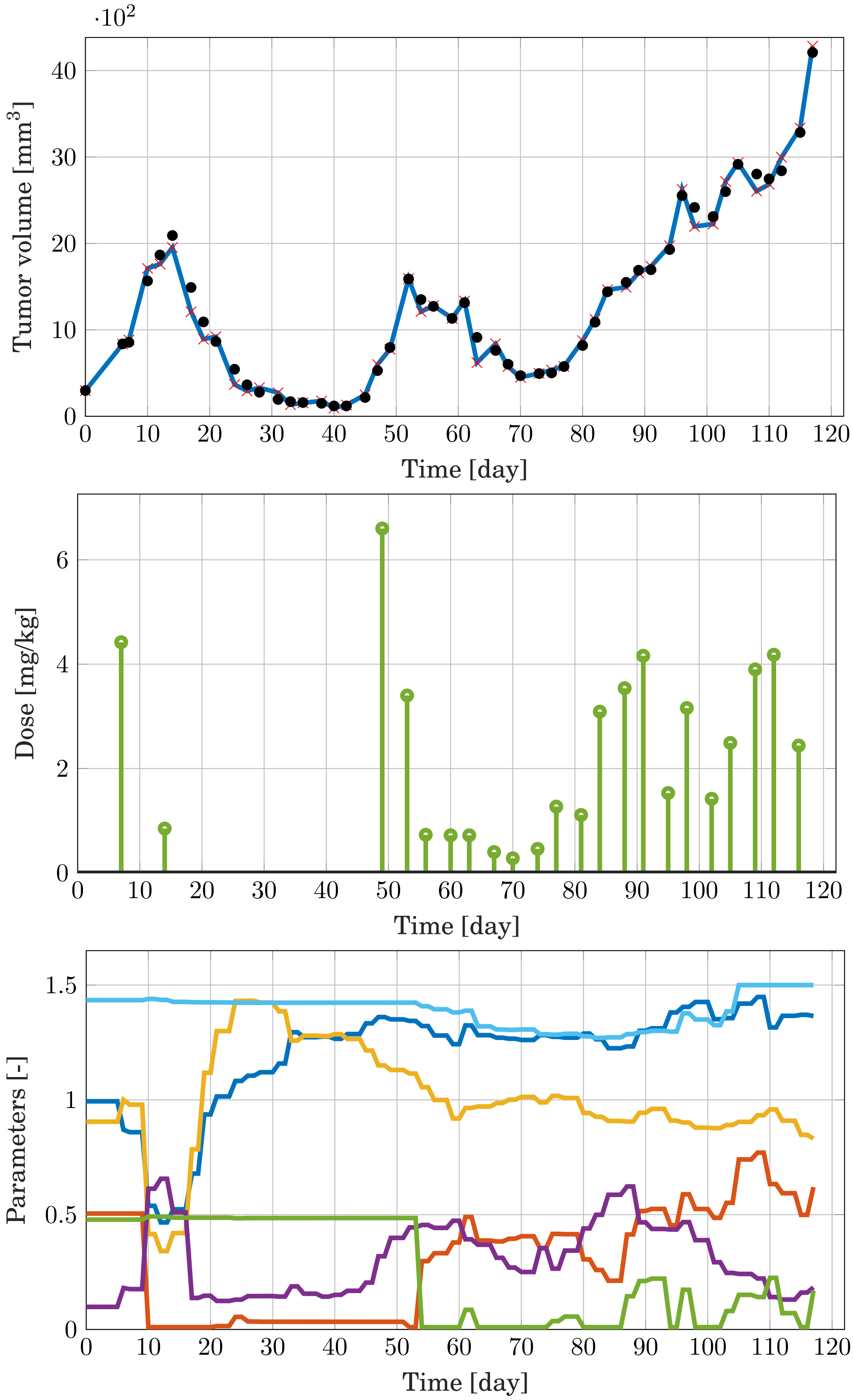


Figure B.38: Evolution of the tumor volume, the corresponding doses and the identified time-varying parameters for $S_{3,18}$. The red crosses represent the tumor volume measurements, while the black dots show their estimations, produced by the MHE. For the parameters a, b, n, w, ED_{50} and c in the bottom figure, the colors blue, red, yellow, purple, green, and turquoise are assigned, respectively.

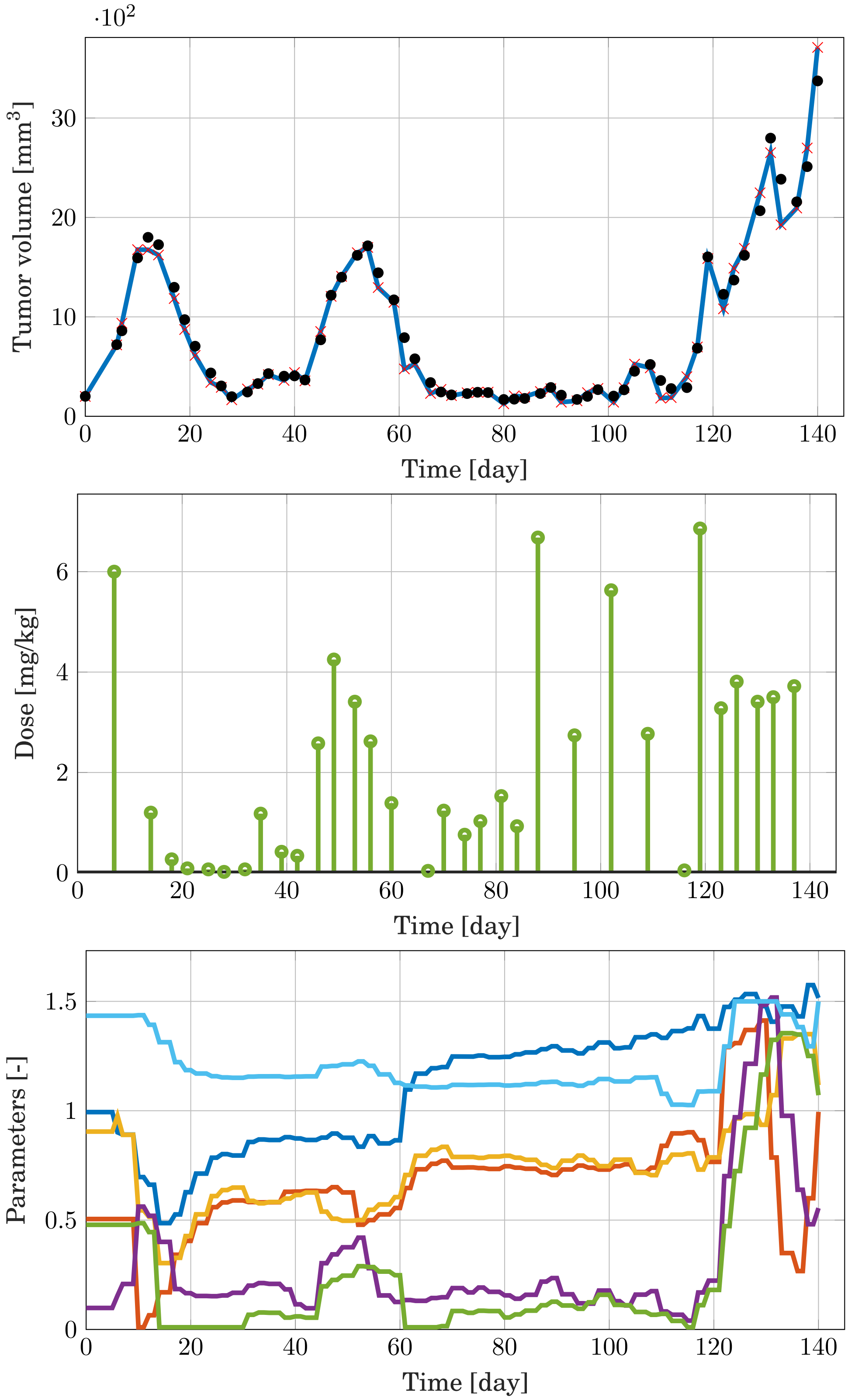


Figure B.39: Evolution of the tumor volume, the corresponding doses and the identified time-varying parameters for $S_{3,20}$. The red crosses represent the tumor volume measurements, while the black dots show their estimations, produced by the MHE. For the parameters a, b, n, w, ED_{50} and c in the bottom figure, the colors blue, red, yellow, purple, green, and turquoise are assigned, respectively.

Bibliography

[R1] M. Dalmartello et al. “European cancer mortality predictions for the year 2022 with focus on ovarian cancer”. In: *Annals of Oncology* 33.3 (Mar. 2022), pp. 330–339. DOI: 10.1016/j.annonc.2021.12.007. URL: https://doi.org/10.1016%5C%2Fj.annonc.2021.12.007.

[R2] Rituparna Maiti. “Metronomic chemotherapy”. In: *Journal of Pharmacology and Pharmacotherapeutics* 5.3 (Sept. 2014), pp. 186–192. DOI: 10.4103/0976-500x.136098. URL: https://doi.org/10.4103/0976-500x.136098.

[R3] Cécile Carrère. “Optimization of an in vitro chemotherapy to avoid resistant tumours”. In: *Journal of Theoretical Biology* 413 (Jan. 2017), pp. 24–33. DOI: 10.1016/j.jtbi.2016.11.009. URL: https://doi.org/10.1016/j.jtbi.2016.11.009.

[R4] Urszula Ledzewicz, Behrooz Amini, and Heinz Schättler. “Dynamics and control of a mathematical model for metronomic chemotherapy”. In: *Mathematical Biosciences and Engineering* 12.6 (Aug. 2015), pp. 1257–1275. DOI: 10.3934/mbe.2015.12.1257. URL: https://doi.org/10.3934/mbe.2015.12.1257.

[R5] J. Aroesty et al. “Tumor growth and chemotherapy: Mathematical methods, computer simulations, and experimental foundations”. In: *Mathematical Biosciences* 17.3-4 (Aug. 1973), pp. 243–300. DOI: 10.1016/0025-5564(73)90072-2. URL: https://doi.org/10.1016%5C%2F0025-5564%5C%2873%5C%2990072-2.

[R6] S. Zietz and C. Nicolini. “Mathematical approaches to optimization of cancer chemotherapy”. In: *Bulletin of Mathematical Biology* 41.3 (1979), pp. 305–324. DOI: 10.1016/s0092-8240(79)90014-4. URL: https://doi.org/10.1016%5C%2Fs0092-8240%5C%2879%5C%2990014-4.

[R7] George W. Swan. “Role of optimal control theory in cancer chemotherapy”. In: *Mathematical Biosciences* 101.2 (Oct. 1990), pp. 237–284. DOI: 10.1016/0025-5564(90)90021-p. URL: https://doi.org/10.1016/0025-5564(90)90021-p.

[R8] Jinghua Shi et al. “A survey of optimization models on cancer chemotherapy treatment planning”. In: *Annals of Operations Research* 221.1 (Mar. 2011), pp. 331–356. DOI: 10.1007/s10479-011-0869-4. URL: https://doi.org/10.1007/s10479-011-0869-4.

[R9] Hoda Sbeity. “Review of Optimization Methods for Cancer Chemotherapy Treatment Planning”. In: *Journal of Computer Science and Systems Biology* 8.2 (2015). DOI: 10.4172/jcsb.1000173. URL: https://doi.org/10.4172/jcsb.1000173.

[R10] Tao Chen, Norman F. Kirkby, and Raj Jena. “Optimal dosing of cancer chemotherapy using model predictive control and moving horizon state/parameter estimation”. In: *Computer Methods and Programs in Biomedicine* 108.3 (Dec. 2012), pp. 973–983. DOI: 10.1016/j.cmpb.2012.05.011. URL: https://doi.org/10.1016/j.cmpb.2012.05.011.

[R11] João P. Belfo and João M. Lemos. *Optimal Impulsive Control for Cancer Therapy*. Springer International Publishing, 2021. DOI: 10.1007/978-3-030-50488-5. URL: https://doi.org/10.1007/978-3-030-50488-5.

[R12] Jozsef K. Tar and Imre J. Rudas. “Geometric Approach to Nonlinear Adaptive Control”. In: *2007 4th International Symposium on Applied Computational Intelligence and Informatics*. IEEE, May 2007. DOI: 10.1109/saci.2007.375477.

[R13] Jozsef K. Tar et al. “Possible adaptive control by tangent hyperbolic fixed point transformations used for controlling the -6-type van der pol oscillator”. In: *2008 IEEE International Conference on Computational Cybernetics*. IEEE, Nov. 2008. DOI: 10.1109/icccyb.2008.4721371.

[R14] Jozsef K. Tar et al. “Preliminary sketch of possible Fixed Point transformations for use in adaptive control”. In: *2008 6th International Symposium on Intelligent Systems and Informatics*. IEEE, Sept. 2008. DOI: 10.1109/sisy.2008.4664920.

[R15] J. K. Tar, I. J. Rudas, and J. F. Bito. “Fixed point stabilization in a novel MRAC control for MIMO systems”. In: *IEEE 8th International Symposium on Intelligent Systems and Informatics*. IEEE, Sept. 2010. DOI: 10.1109/sisy.2010.5647393.

[R16] Adrienn Dineva et al. “Replacement of parameter tuning with simple calculation in adaptive control using Sigmoid generated fixed point transformation”. In: *2015 IEEE 13th International Symposium on Intelligent Systems and Informatics (SISY)*. IEEE, Sept. 2015. DOI: 10.1109/sisy.2015.7325374.

[R17] J. K. Tar et al. “Convergence stabilization by parameter tuning in Robust Fixed Point Transformation based adaptive control of underactuated MIMO systems”. In: *2010 International Joint Conference on Computational Cybernetics and Technical Informatics*. IEEE, May 2010. DOI: 10.1109/icccyb.2010.5491239.

[R18] Adrienn Dineva et al. “Generalization of a sigmoid generated Fixed Point Transformation from SISO to MIMO systems”. In: *2015 IEEE 19th International Conference on Intelligent Engineering Systems (INES)*. IEEE, Sept. 2015. DOI: 10.1109/ines.2015.7329694.

[R19] Adrienn Dineva et al. “Adaptive control of underactuated mechanical systems using improved "Sigmoid Generated Fixed Point Transformation" and scheduling strategy”. In: *2016 IEEE 14th International Symposium on Applied Machine Intelligence and Informatics (SAMI)*. IEEE, Jan. 2016. DOI: 10.1109/sami.2016.7423006.

[R20] Bence G. Czako and Krisztian Kosi. “Novel method for quadcopter controlling using nonlinear adaptive control based on robust fixed point transformation phenomena”. In: *2017 IEEE 15th International Symposium on Applied Machine Intelligence and Informatics (SAMI)*. IEEE, Jan. 2017. DOI: 10.1109/sami.2017.7880320.

[R21] Levente Kovács et al. “Robust Fixed Point Transformation based Proportional-Derivative Control of Angiogenic Tumor Growth”. In: *IFAC-PapersOnLine* 51.4 (2018), pp. 894–899. DOI: 10.1016/j.ifacol.2018.06.110.

[R22] Árpád Varga et al. “Experimental and Simulation-Based Performance Analysis of a Computed Torque Control (CTC) Method Running on a Double Rotor Aeromechanical Testbed”. In: *Electronics* 10.14 (July 2021), p. 1745. DOI: 10.3390/electronics10141745.

[R23] F. Allgower et al. “Nonlinear Predictive Control and Moving Horizon Estimation - An Introductory Overview”. In: *Advances in Control*. Springer London, 1999, pp. 391–449. DOI: 10.1007/978-1-4471-0853-5_19.

[R24] M. Diehl et al. “Fast Direct Multiple Shooting Algorithms for Optimal Robot Control”. In: *Lecture Notes in Control and Information Sciences*. Springer Berlin Heidelberg, 2005, pp. 65–93. DOI: 10.1007/978-3-540-36119-0_4.

[R25] András Füredi et al. “Pegylated liposomal formulation of doxorubicin overcomes drug resistance in a genetically engineered mouse model of breast cancer”. In: *Journal of Controlled Release* 261 (Sept. 2017), pp. 287–296. DOI: 10.1016/j.jconrel.2017.07.010. URL: https://doi.org/10.1016/j.jconrel.2017.07.010.

[R26] Philipp M. Altrock, Lin L. Liu, and Franziska Michor. “The mathematics of cancer: integrating quantitative models”. In: *Nature Reviews Cancer* 15.12 (Nov. 2015), pp. 730–745. DOI: 10.1038/nrc4029. URL: https://doi.org/10.1038/nrc4029.

[R27] Christophe Deroulers et al. “Modeling tumor cell migration: From microscopic to macroscopic models”. In: *Phys. Rev. E* 79 (3 Mar. 2009), p. 031917. DOI: 10.1103/PhysRevE.79.031917. URL: https://link.aps.org/doi/10.1103/PhysRevE.79.031917.

[R28] Ali Masoudi-Nejad et al. “Cancer systems biology and modeling: Microscopic scale and multiscale approaches”. In: *Seminars in Cancer Biology* 30 (2015). Cancer modeling and network biology, pp. 60–69. ISSN: 1044-579X. DOI: https://doi.org/10.1016/j.semcancer.2014.03.003. URL: https://www.sciencedirect.com/science/article/pii/S1044579X1400039X.

[R29] X. Liu et al. “Somatic loss of BRCA1 and p53 in mice induces mammary tumors with features of human BRCA1-mutated basal-like breast cancer”. In: *Proceedings of the National Academy of Sciences* 104.29 (July 2007), pp. 12111–12116. DOI: 10.1073/pnas.0702969104. URL: https://doi.org/10.1073/pnas.0702969104.

[R30] Lilla Hámori et al. “Establishment and Characterization of a Brca1-/-, p53-/- Mouse Mammary Tumor Cell Line”. In: *International Journal of Molecular Sciences* 21.4 (Feb. 2020), p. 1185. DOI: 10.3390/ijms21041185. URL: https://doi.org/10.3390/ijms21041185.

[R31] S. Rottenberg et al. “Selective induction of chemotherapy resistance of mammary tumors in a conditional mouse model for hereditary breast cancer”. In: *Proceedings of the National Academy of Sciences* 104.29 (July 2007), pp. 12117–12122. DOI: 10.1073/pnas.0702955104. URL: https://doi.org/10.1073/pnas.0702955104.

[R32] Edina Karai et al. “Celecoxib Prevents Doxorubicin-Induced Multidrug Resistance in Canine and Mouse Lymphoma Cell Lines”. In: *Cancers* 12.5 (Apr. 2020), p. 1117. DOI: 10.3390/cancers12051117. URL: https://doi.org/10.3390/cancers12051117.

[R33] András Füredi et al. “Identification and Validation of Compounds Selectively Killing Resistant Cancer: Delineating Cell LineSpecific Effects from P-GlycoproteinInduced Toxicity”. In: *Molecular Cancer Therapeutics* 16.1 (Oct. 2016), pp. 45–56. DOI: 10.1158/1535-7163.mct-16-0333-t. URL: https://doi.org/10.1158/1535-7163.mct-16-0333-t.

[R34] Zita Rádai et al. “Synthesis and anticancer cytotoxicity with structural context of an α-hydroxyphosphonate based compound library derived from substituted benzaldehydes”. In: *New Journal of Chemistry* 43.35 (2019), pp. 14028–14035. DOI: 10.1039/c9nj02144b. URL: https://doi.org/10.1039/c9nj02144b.

[R35] Johanna Sápi et al. "Tumor Volume Estimation and Quasi-Continuous Administration for Most Effective Bevacizumab Therapy". In: *PLOS ONE* 10.11 (Nov. 2015). Ed. by Francesco Bertolini, e0142190. DOI: 10.1371/journal.pone.0142190.

[R36] Dániel András Drexler et al. "Experimental data-driven tumor modeling for chemotherapy". In: *IFAC-PapersOnLine* 53.2 (2020), pp. 16245–16250. DOI: 10.1016/j.ifacol.2020.12.619. URL: https://doi.org/10.1016/j.ifacol.2020.12.619.

[R37] Dániel András Drexler et al. "Tumor dynamics modeling based on formal reaction kinetics". In: *Acta Polytechnica Hungarica* 16.10 (2019), pp. 31–44. DOI: 10.12700/aph.16.10.2019.10.3. URL: https://doi.org/10.12700/aph.16.10.2019.10.3.

[R38] A. M. Samoilenko and N. A. Perestyuk. *Impulsive Differential Equations*. WORLD SCIENTIFIC, Aug. 1995. DOI: 10.1142/2892.

[R39] H. K. Khalil. *Nonlinear systems*. Upper Saddle River, N.J: Prentice Hall, 2002. ISBN: 978-0130673893.

[R40] Alberto Isidori. "The zero dynamics of a nonlinear system: From the origin to the latest progresses of a long successful story". en. In: *Eur. J. Control* 19.5 (Sept. 2013), pp. 369–378.

[R41] Eberhard Zeidler. *Nonlinear Functional Analysis and its Applications I: Fixed-Point Theorems*. Springer-Verlag, 1986. ISBN: 978-0-387-90914-1.

[R42] Masoud Soroush and Costas Kravaris. "Discrete-time nonlinear controller synthesis by input/output linearization". In: *AIChE J.* 38.12 (Dec. 1992), pp. 1923–1945.

[R43] Bertalan Csanadi et al. "Revisiting Lyapunov's Technique in the Fixed Point Transformation-Based Adaptive Control". In: *2018 IEEE 22nd International Conference on Intelligent Engineering Systems (INES)*. IEEE, June 2018. DOI: 10.1109/ines.2018.8523923. URL: https://doi.org/10.1109/ines.2018.8523923.

[R44] József K. Tar et al. "Robust fixed point transformations in adaptive control using local basin of attraction". English. In: *Acta Polytechnica Hungarica* 6.1 (Dec. 2009), pp. 21–38. ISSN: 1785-8860.

[R45] Michael A. Henson and Dale E. Seborg, eds. *Nonlinear Process Control*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1997. ISBN: 0-13-625179-X.

[R46] Krisztian Kosi et al. "Chaos formation and reduction in robust fixed point transformations based adaptive control". In: *2012 IEEE 4th International Conference on Nonlinear Science and Complexity (NSC)*. Budapest, Hungary: IEEE, Aug. 2012.

[R47] Katsuhiko Ogata. *Discrete-Time Control Systems (2nd Ed.)* USA: Prentice-Hall, Inc., 1995. ISBN: 0130342815.

[R48] Mate Siket, Gyorgy Eigner, and Levente Kovacs. "Sensitivity and identifiability analysis of a third-order tumor growth model". In: *2020 IEEE 15th International Conference of System of Systems Engineering (SoSE)*. IEEE, June 2020. DOI: 10.1109/sose50414.2020.9130530. URL: https://doi.org/10.1109/sose50414.2020.9130530.

[R49] Oana Chi, Julio R. Banga, and Eva Balsa-Canto. "GenSSI: a software toolbox for structural identifiability analysis of biological models". In: *Bioinformatics* 27.18 (July 2011), pp. 2610–2611. DOI: 10.1093/bioinformatics/btr431. URL: https://doi.org/10.1093/bioinformatics/btr431.

[R50] Eva Balsa-Canto et al. “AMIGO2, a toolbox for dynamic modeling, optimization and control in systems biology”. In: *Bioinformatics* 32.21 (July 2016), pp. 3357–3359. DOI: 10.1093/bioinformatics/btw411. URL: https://doi.org/10.1093/bioinformatics/btw411.

[R51] Máté Siket et al. “State and Parameter Estimation of a Mathematical Carcinoma Model under Chemotherapeutic Treatment”. In: *Applied Sciences* 10.24 (Dec. 2020), p. 9046. DOI: 10.3390/app10249046.

[R52] Oana-Teodora Chis, Julio R. Banga, and Eva Balsa-Canto. “Structural Identifiability of Systems Biology Models: A Critical Comparison of Methods”. In: *PLoS ONE* 6.11 (Nov. 2011). Ed. by Johannes Jaeger, e27755. DOI: 10.1371/journal.pone.0027755. URL: https://doi.org/10.1371/journal.pone.0027755.

[R53] I.A. Hiskens and M.A. Pai. “Trajectory sensitivity analysis of hybrid systems”. In: *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications* 47.2 (2000), pp. 204–220. DOI: 10.1109/81.828574. URL: https://doi.org/10.1109/81.828574.

[R54] Sijia Geng and Ian A. Hiskens. “Second-Order Trajectory Sensitivity Analysis of Hybrid Systems”. In: *IEEE Transactions on Circuits and Systems I: Regular Papers* 66.5 (May 2019), pp. 1922–1934. DOI: 10.1109/tcsi.2019.2903196. URL: https://doi.org/10.1109/tcsi.2019.2903196.

[R55] Hongtu Zhu, Minggao Gu, and Bradley Peterson. “Maximum likelihood from spatial random effects models via the stochastic approximation expectation maximization algorithm”. In: *Statistics and Computing* 17.2 (Jan. 2007), pp. 163–177. DOI: 10.1007/s11222-006-9012-9. URL: https://doi.org/10.1007/s11222-006-9012-9.

[R56] M L Lindstrom and D M Bates. “Nonlinear mixed effects models for repeated measures data”. In: *Biometrics* 46.3 (Sept. 1990), pp. 673–687.

[R57] Kenneth R. Muske and James B. Rawlings. “Nonlinear Moving Horizon State Estimation”. In: *Methods of Model Based Process Control*. Springer Netherlands, 1995, pp. 349–365. DOI: 10.1007/978-94-011-0135-6_14. URL: https://doi.org/10.1007/978-94-011-0135-6_14.

[R58] Angelo Alessandri et al. “Advances in moving horizon estimation for nonlinear systems”. In: *49th IEEE Conference on Decision and Control (CDC)*. IEEE, Dec. 2010. DOI: 10.1109/cdc.2010.5718126. URL: https://doi.org/10.1109/cdc.2010.5718126.

[R59] Lars Grüne and Jürgen Pannek. *Nonlinear Model Predictive Control*. Springer International Publishing, 2017. DOI: 10.1007/978-3-319-46024-6. URL: https://doi.org/10.1007/978-3-319-46024-6.

[R60] S. Joe Qin and Thomas A. Badgwell. “An Overview of Nonlinear Model Predictive Control Applications”. In: *Nonlinear Model Predictive Control*. Birkhäuser Basel, 2000, pp. 369–392. DOI: 10.1007/978-3-0348-8407-5_21. URL: https://doi.org/10.1007/978-3-0348-8407-5_21.

[R61] Ying Ding et al. “Model predictive control and its application in agriculture: A review”. In: *Computers and Electronics in Agriculture* 151 (Aug. 2018), pp. 104–117. DOI: 10.1016/j.compag.2018.06.004. URL: https://doi.org/10.1016/j.compag.2018.06.004.

[R62] Honggui Han and Junfei Qiao. “Nonlinear Model-Predictive Control for Industrial Processes: An Application to Wastewater Treatment Process”. In: *IEEE Transactions on Industrial Electronics* 61.4 (Apr. 2014), pp. 1970–1982. DOI: 10.1109/tie.2013.2266086. URL: https://doi.org/10.1109/tie.2013.2266086.

[R63] E. F. Camacho and C. Bordons. *Model Predictive control*. Springer London, 2007. DOI: 10.1007/978-0-85729-398-5. URL: https://doi.org/10.1007/978-0-85729-398-5.

[R64] Achille Messac. *Optimization in Practice with MATLAB: For Engineering Students and Professionals*. Cambridge University Press, 2015. DOI: 10.1017/CBO9781316271391.

[R65] H. J. Hedrich and G. Bullock. *The Laboratory Mouse*. Elsevier, 2004. DOI: 10.1016/b978-0-12-336425-8.x5051-1.

[R66] Jason T. Rich et al. “A practical guide to understanding Kaplan-Meier curves”. In: *OtolaryngologyHead and Neck Surgery* 143.3 (Sept. 2010), pp. 331–336. DOI: 10.1016/j.otohns.2010.05.007. URL: https://doi.org/10.1016/j.otohns.2010.05.007.

[R67] J Martin Bland and Douglas G Altman. “The logrank test”. In: *BMJ* 328.7447 (Apr. 2004), p. 1073. DOI: 10.1136/bmj.328.7447.1073. URL: https://doi.org/10.1136/bmj.328.7447.1073.

[R68] Nicolas André, Manon Carré, and Eddy Pasquier. “Metronomics: towards personalized chemotherapy?” In: *Nature Reviews Clinical Oncology* 11.7 (June 2014), pp. 413–431. DOI: 10.1038/nrclinonc.2014.89. URL: https://doi.org/10.1038/nrclinonc.2014.89.

[R69] Daniel Andras Drexler et al. “Modeling of tumor growth incorporating the effect of pegylated liposomal doxorubicin”. In: *2019 IEEE 23rd International Conference on Intelligent Engineering Systems (INES)*. IEEE, Apr. 2019. DOI: 10.1109/ines46365.2019.9109532. URL: https://doi.org/10.1109/ines46365.2019.9109532.

[R70] Daniel Andras Drexler et al. “Comparison of Michaelis-Menten kinetics modeling alternatives in cancer chemotherapy modeling”. In: *2019 IEEE 13th International Symposium on Applied Computational Intelligence and Informatics (SACI)*. IEEE, May 2019. DOI: 10.1109/saci46893.2019.9111543. URL: https://doi.org/10.1109/saci46893.2019.9111543.

[R71] Krisztian Kosi, Jozsef K. Tar, and Imre J. Rudas. “Improvement of the stability of RFPT-based adaptive controllers by observing precursor oscillations”. In: *2013 IEEE 9th International Conference on Computational Cybernetics (ICCC)*. IEEE, July 2013. DOI: 10.1109/icccyb.2013.6617601.

[R72] Jozsef K. Tar and Imre J. Rudas. “Analysis of the Fixed Point Transformation Based Adapive Robot Control”. In: *2008 International Conference on Intelligent Engineering Systems*. IEEE, Feb. 2008. DOI: 10.1109/ines.2008.4481264.

[R73] Arpad Varga, Gyorgy Eigner, and JOzsef K. Tar. “Simple aeromechanical test bed for preliminary performance evaluation of robust nonlinear control methods”. In: *2018 IEEE 18th International Symposium on Computational Intelligence and Informatics (CINTI)*. IEEE, Nov. 2018. DOI: 10.1109/cinti.2018.8928228.

[R74] Arpad Varga et al. “Fixed Point Iteration-based Adaptive Control for a Delayed Differential Equation Model of Diabetes Mellitus”. In: *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*. IEEE, Oct. 2019. DOI: 10.1109/smc.2019.8914617.

[R75] Levente Kovacs et al. “An opportunity of using Robust Fixed Point Transformation-based controller design in case of Type 1 Diabetes Mellitus”. In: *2019 First International Conference on Societal Automation (SA)*. IEEE, Sept. 2019. DOI: 10.1109/sa47457.2019.8938069.

[R76] Hamza Khan, Aurel Galantai, and Jozsef K. Tar. “Adaptive solution of the inverse kinematic task by fixed point transformation”. In: *2017 IEEE 15th International Symposium on Applied Machine Intelligence and Informatics (SAMI)*. IEEE, Jan. 2017. DOI: 10.1109/sami.2017.7880312.

[R77] Jozsef K. Tar et al. “Tackling complexity and missing information in adaptive control by fixed point transformation-based approach”. In: *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, Oct. 2016. DOI: 10.1109/smc.2016.7844454. URL: https://doi.org/10.1109/smc.2016.7844454.

[R78] Mouhacine Benosman. “Multi-parametric extremum seeking-based iterative feedback gains tuning for nonlinear control”. In: *International Journal of Robust and Nonlinear Control* 26.18 (Apr. 2016), pp. 4035–4055. DOI: 10.1002/rnc.3547. URL: https://doi.org/10.1002/rnc.3547.

[R79] Yaman Arkun and Jean-Paul Calvet. “Robust stabilization of input/output linearizable systems under uncertainty and disturbances”. In: *AIChE Journal* 38.8 (Aug. 1992), pp. 1145–1156. DOI: 10.1002/aic.690380802. URL: https://doi.org/10.1002/aic.690380802.

[R80] JEAN-JACQUES E. SLOTINE and J. KARL HEDRICK. “Robust input-output feedback linearization”. In: *International Journal of Control* 57.5 (May 1993), pp. 1133–1139. DOI: 10.1080/00207179308934435. URL: https://doi.org/10.1080/00207179308934435.

[R81] Daniel Andras Drexler, Johanna Sapi, and Levente Kovacs. “A minimal model of tumor growth with angiogenic inhibition using bevacizumab”. In: *2017 IEEE 15th International Symposium on Applied Machine Intelligence and Informatics (SAMI)*. IEEE, Jan. 2017. DOI: 10.1109/sami.2017.7880300. URL: https://doi.org/10.1109/sami.2017.7880300.

[R82] Dániel András Drexler, Johanna Sápi, and Levente Kovács. “Positive nonlinear control of tumor growth using angiogenic inhibition”. In: *IFAC-PapersOnLine* 50.1 (July 2017), pp. 15068–15073.

Own Publications Pertaining to Theses

[C1] Bence Géza Czakó, Dániel András Drexler, and Levente Kovács. “Discrete time derivation of the Robust Fixed-Point Transformation method”. In: *IFAC-PapersOnLine* 55.1 (2022), pp. 535–540.

[C2] Bence Geza Czako, Daniel Andras Drexler, and Levente Kovacs. “Continuous time Robust Fixed Point Transformations based control”. In: *2019 IEEE AFRICON*. IEEE, Sept. 2019. DOI: 10.1109/africon46755.2019.9133741. URL: https://doi.org/10.1109/africon46755.2019.9133741.

[C3] Bence Czakó and Levente Kovács. “Nonlinear Model Predictive Control Using Robust Fixed Point Transformation-Based Phenomena for Controlling Tumor Growth”. In: *Machines* 6.4 (Oct. 2018), p. 49. DOI: 10.3390/machines6040049. URL: https://doi.org/10.3390/machines6040049.

[C4] Bence Czako, Johanna Sapi, and Levente Kovacs. “Model-based optimal control method for cancer treatment using model predictive control and robust fixed point method”. In: *2017 IEEE 21st International Conference on Intelligent Engineering Systems (INES)*. IEEE, Oct. 2017. DOI: 10.1109/ines.2017.8118569. URL: https://doi.org/10.1109/ines.2017.8118569.

[C5] Bence Geza Czako, Daniel Andras Drexler, and Levente Kovacs. “Impulsive Control of Tumor Growth via Nonlinear Model Predictive Control Using Direct Multiple Shooting”. In: *2020 European Control Conference (ECC)*. IEEE, May 2020. DOI: 10.23919/ecc51009.2020.9143755. URL: https://doi.org/10.23919/ecc51009.2020.9143755.

[C6] Levente Kovács et al. “Experimental Closed-Loop Control of Breast Cancer in Mice”. In: *Complexity* 2022 (May 2022). Ed. by Qingling Wang, pp. 1–10. DOI: 10.1155/2022/9348166. URL: https://doi.org/10.1155/2022/9348166.

[C7] Bence Czakó et al. “Chemotherapy Optimization using Moving Horizon Estimation based Nonlinear Model Predictive Control”. In: *IFAC-PapersOnLine* 54.15 (2021), pp. 215–220.

[C8] Bence Czakó, Dániel András Drexler, and Levente Kovács. “Time-Varying Parameter Identification of a Tumor Growth Model Using Moving Horizon Estimation”. In: *IEEE 26th International Conference on Intelligent Engineering Systems (INES 2022) - Manuscript submitted for publication* (2022).

[C9] Levente Kovacs et al. “An opportunity of using Robust Fixed Point Transformation-based controller design in case of Type 1 Diabetes Mellitus”. In: *2019 First International Conference on Societal Automation (SA)*. IEEE, Sept. 2019.

[C10] Bence Geza Czako, Daniel Andras Drexler, and Levente Kovacs. “Control of a T1DM model using Robust Fixed-Point Transformations based control with disturbance rejection”. In: *IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR) - Manuscript submitted for publication*. IEEE, May 2022.

[C11] Bence G. Czako and Krisztian Kosi. “Novel method for quadcopter controlling using nonlinear adaptive control based on robust fixed point transformation phenomena”. In: *2017 IEEE 15th International Symposium on Applied Machine Intelligence and Informatics (SAMI)*. IEEE, Jan. 2017. DOI: 10.1109/sami.2017.7880320.

Own Publications Not Pertaining to Theses

[Cx1] Bence G. Czako and Krisztian Kosi. “Novel method for quadcopter controlling using nonlinear adaptive control based on robust fixed point transformation phenomena”. In: *2017 IEEE 15th International Symposium on Applied Machine Intelligence and Informatics (SAMI)*. IEEE, Jan. 2017. DOI: 10.1109/sami.2017.7880320. URL: https://doi.org/10.1109/sami.2017.7880320.

[Cx2] Bence Czako, Johanna Sapi, and Levente Kovacs. “Model-based optimal control method for cancer treatment using model predictive control and robust fixed point method”. In: *2017 IEEE 21st International Conference on Intelligent Engineering Systems (INES)*. IEEE, Oct. 2017. DOI: 10.1109/ines.2017.8118569. URL: https://doi.org/10.1109/ines.2017.8118569.

[Cx3] Levente Kovacs et al. “An opportunity of using Robust Fixed Point Transformation-based controller design in case of Type 1 Diabetes Mellitus”. In: *2019 First International Conference on Societal Automation (SA)*. IEEE, Sept. 2019.

[Cx4] Bence Geza Czako, Daniel Andras Drexler, and Levente Kovacs. “Control of a T1DM model using Robust Fixed-Point Transformations based control with disturbance rejection”. In: *IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR) - Manuscript submitted for publication*. IEEE, May 2022.