

Tumor model parameter estimation for therapy optimization using artificial neural networks

Melánia Puskás^{1,2} and Dániel András Drexler¹

Abstract—Therapy optimization and personalization in cancer treatment requires reliable mathematical models. A key issue in personalization is the identification of the model parameters. We employ artificial neural networks to identify the model parameters based on few measurements using a priori information about the range of the parameters. The training data are generated in silico on known parameter intervals, taking into consideration the experimental setup we use to validate our results. The estimated parameters can be used to track the change of the parameters and can also be used as initial guesses for identification algorithms using local search.

I. INTRODUCTION

A promising combination of engineering and medical knowledge is the optimization and personalization of drug delivery, like e.g., tumor chemotherapy. Optimization of cancer chemotherapy can minimize the injection doses, thus side effects and treatment costs, and may also prevent the development of drug resistance [1]–[4]. There can be various methods to solve the therapy optimization problem, one can use the results of impulsive control [4]–[7], which adapts to the specific properties of injections, while classical control theory can also be deployed [8]–[10]. We consider an optimization problem with an impulsive tumor model [11], [12], and concentrate on the parameter estimation whose result can be used by the therapy optimization algorithm.

Optimization requires a reliable mathematical model of the tumor dynamics and the pharmacodynamic/pharmacokinetic (PDPK) model of the drug. The number of tumor models available in the literature is high, see e.g., the review done by Jarrett et al. [13] or Altrock et al. [14]. There are models focusing on tumor growth and supportive vasculature growth [15], some focusing on immune cells and healthy cells as well [16], also some models incorporate drug resistance [1]. In our work, we use a mathematical model based on formal

reaction kinetics. The model captures some crucial physiological phenomena, like tumor cell proliferation, tumor cell necrosis, dead tumor cell washout, and the PDPK model of the chemotherapeutic agent [2], [17], discussed in Subsection II-A.

Besides a reliable tumor dynamics and PDPK model, the model parameters for the specific patient, tumor and drug are also crucial for therapy optimization and personalization. Simple models have relatively few parameters, however, their modeling power may not be satisfactory. More complex models can be used more efficiently for therapy optimization, however, they have more parameters which makes identification harder. It can also occur that some parameters are not identifiable based on the measurements from a given experimental setup, or the sensitivity of some parameters is low, like it has been analysed by Siket et al. [18], [19]. If we have a priori knowledge about the model parameters, e.g., their range, we can use this information to generate measurement data in silico and train neural networks for the parameter estimation problem. The a priori information can be acquired from real measurements using mixed-effect models [2], [20], [21]. The disadvantage of the estimation with mixed-effect models is that they use local algorithm and highly depend on the initial estimation of the parameters.

Soft computing techniques are often used for parameter estimation, see e.g., a comparison in the work of Cakir et. al [22] or Chatuverdi and Singh [23]. The parameter estimation of a Lotka-Volterra model using particle swarm optimization is provided by Lazzús et. al [24]. Parameters of nonlinear MIMO dynamic systems are estimated using neural computing by Liu et. al [25]. Application of soft computing techniques is useful, since the drawbacks of conventional identification techniques, like getting stuck in local minima, are avoided. However, they require a priori knowledge about the values of the parameters.

We use neural networks to provide a good initial estimation of the model parameters [26], which provide estimation based on noisy measurements. We use four-layered, feed-forward architecture, detailed in Subsection II-B. Two different neural networks are trained, one is used to estimate the pharmacodynamic and the pharmacokinetic parameters (we will call this network the PDPK network). The other one is used to estimate the tumor growth parameters. We will call this network the TDNN (tumor dynamics neural network). The inputs of the neural networks are the tumor volume measured every day for a week except the weekends and injections given two times a week based on the experimental setup discussed in Section III. The networks are trained based

*This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 679681). Project no. 2019-1.3.1-KK-2019-00007. has been implemented with the support provided from the National Research, Development and Innovation Fund of Hungary, financed under the 2019-1.3.1-KK funding scheme. The present work has also been supported by the Hungarian National Research, Development and Innovation Office (2018- 2.1.11-T ÉT-SI-2018-00007 and SNN 125739). The research was also supported by the Eötvös Loránd Research Network Secretariat under grant agreement no. ELKH KÖ-40/2020 ('Development of cyber-medical systems based on AI and hybrid cloud methods').

¹The authors are with the Physiological Research Center, Research and Innovation Center of Óbuda University, Óbuda University, Budapest, Hungary {puskas.melania,drexler.daniel}@uni-obuda.hu

²Melánia Puskás is also with the Institute for Computer Science and Control (SZTAKI), Eötvös Loránd Research Network (ELKH), Budapest, Hungary

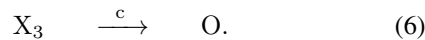
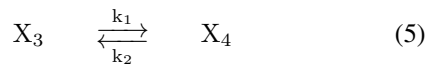
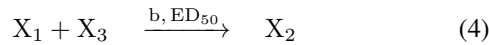
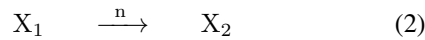
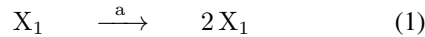
on data generated in silico on a large span of parameters. The estimation is carried out two times a week, before the injection is given, which uses the model parameters given by the neural network estimator combined with a local search algorithm.

The challenge in the application of soft computing techniques for physiological problems is that the amount of measurement data is low, and not sufficient for training. We provide an algorithm to generate training data for the neural network which can be later applied for the measurements from the real scenario. The training data are generated on a large span of parameters in order to enhance the operation of local estimators effectively. Although, due to the large span of the training data and the high dimension of the problem, an efficient operation requires extremely large number of training data sampled from the span of the parameters. We carry out training data generation with randomized parameters on a relatively small number of virtual data and show that we can get good initial estimation for the parameters with large probability in Subsections IV-A and IV-B.

II. PRELIMINARIES

A. Tumor model

We use a tumor model described by a fourth-order system of differential equations [17]. In order to get an easily interpretable model, we use formal reaction kinetics to describe the modeled phenomena: let X_1 , X_2 , X_3 and X_4 be the fictive species corresponding to the living tumor volume, dead tumor volume, drug level in central compartment and drug level in peripheral compartment, respectively. Then the formal reaction kinetic equations are



where (1) is tumor proliferation with rate a , (2) is the necrosis of living tumor cells independent of the drug with rate n , (3) is the washout (absorption) of the dead tumor cells with washout rate w , (4) is the pharmacodynamics of the drug with maximal effect rate b and median effective dose ED_{50} , (5) is the drug transport between the central and peripheral compartment with rate constants k_1 and k_2 , while (6) is the depletion of the drug with clearance c .

In order to create the differential equations corresponding to the fictive reaction (1)-(6), first we create the velocities corresponding to each reaction step. One can consider a reaction step with mass action kinetics, this case the velocity will be linear in all the species on the left-hand side of the reaction step (these species are called the reactant species). This may result in a multilinear velocity if there are more reactant species in the reaction step. This velocity has no

upper bound, due to the linearity. One can also consider Michealis-Menten kinetics, in this case, the velocity is a Hill function of the reactant species, which models that there is an upper limit for the velocity of the reaction step. One can also consider mixture of mass action kinetics and Michaelis-Menten kinetics for the different reactant species in the given reaction step. The velocities are added to the differential equation of a given species with a positive sign if the species is produced during the reaction step, and added with a negative sign, if the species is consumed in the reaction step. For more details, the reader should consult [27], [28]. Here, we consider the reaction steps (1)-(3) and (5)-(6) with mass-action kinetics, and the reaction step (4) with mass-action kinetics in the species X_1 with rate b and Michaelis-Menten kinetics in the species X_3 with Michaelis-Menten constant ED_{50} , and get the system of differential equations

$$\dot{x}_1 = (a - n)x_1 - b \frac{x_1 x_3}{ED_{50} + x_3} \quad (7)$$

$$\dot{x}_2 = nx_1 + b \frac{x_1 x_3}{ED_{50} + x_3} - wx_2 \quad (8)$$

$$\dot{x}_3 = -(c + k_1)x_3 + k_2 x_4 - b_k \frac{x_1 x_3}{ED_{50} + x_3} \quad (9)$$

$$\dot{x}_4 = k_1 x_3 - k_2 x_4. \quad (10)$$

The time function x_1 corresponds to the living tumor volume [mm^3], x_2 is the time function of the dead tumor volume [mm^3], x_3 is the time function of the drug serum level in the central compartment [ng/ml], while x_4 is the time function of the drug serum level in the peripheral compartment [ng/ml]. Note that we use a modified rate constant b_k in (9) since the dimension of the drug level is different from the dimension of the tumor volumes.

The input of the system is impulsive effect on the central compartment, which is the injection dose u_k [mg/kg] at time t_k with $u_k \geq 0$ and $t_k \geq 0$, and $k = 1, 2, \dots$. The impulsive effect changes the drug level in the central compartment discontinuously as

$$x_3(t_k^+) = x_3(t_k^-) + \kappa u_k, \quad (11)$$

where κ is a constant to convert the drug dosage to serum level. The output of the system is

$$y = x_1 + x_2, \quad (12)$$

which is the sum of the living and dead tumor volume, that is the measured variable during the experiments.

Our training data generation is based on a priori knowledge acquired during parameter identification using mixed-effect models in our previous research [2], [17]. These identification results showed that $b_k = 0$, thus we will consider this parameter as zero throughout the paper.

B. Neural network

A four-layered feed-forward neural network is used to estimate the model parameters, with sigmoid activation functions in the hidden layers and linear functions in the output layer.

Thus, the r th neuron of the l th layer realizes the function

$$y_r^{(l)}(\mathbf{x}) = \varphi \left(\sum_{i=1}^{N_i^{(l)}} w_{r,i}^{(l)} x_i^{(l)}(p) + \underbrace{w_{r,N_i^{(l)}+1}^{(l)}}_{\text{offset}} \right), \quad (13)$$

where $r = 1, 2, \dots, R^{(l)}$ and $R^{(l)}$ is the number of neurons in the l th hidden layer, $l = 1, 2$, the superscript (l) refers to layer l , $\mathbf{x}^{(l)}$ is the input of the layer l , $x_i^{(l)}$ is the i th input of layer l , where $i = 1, 2, \dots, N_i^{(l)}$ and $N_i^{(l)}$ is the number of inputs of layer l , $w_{r,i}^{(l)}$ is the weight of the i th input of the r th neuron in layer l , and $w_{r,N_i^{(l)}+1}^{(l)}$ is the weight of the bias of the r th neuron in layer l . We use sigmoid activation function

$$\varphi(\epsilon) = \frac{2}{1 + e^{-2\epsilon}} - 1. \quad (14)$$

The m th output of the neural network is

$$y_m^{(3)}(\mathbf{x}) = \sum_{r=1}^{R^{(2)}} w_{m,r}^{(3)} y_r^{(2)}(\mathbf{x}) + \underbrace{w_{m,R^{(2)}+1}^{(3)}}_{\text{offset}}, \quad (15)$$

where $w_{m,r}^{(3)}$ is the weight of the m th neuron and r th input in the output layer, and $w_{m,R^{(2)}+1}^{(3)}$ is the weight of the bias for the m th neuron in the output layer.

We train the neural network with supervised learning using error backpropagation with Levenberg-Marquardt algorithm [29]. The training data are generated in silico as described in Section III.

III. TRAINING DATA

The training data is generated with some randomized parameters from the ranges shown in Tables I and II. These ranges are based on the results of parameter identification carried out with a mixed-effect mode in [17]. The parameters span a large interval, which makes the estimation task hard. There are some parameters whose lower and upper bounds differ in three orders of magnitude, so finding the good initial estimation for a local estimator is not a trivial task. One advantage of using a well trained neural network is that it can provide a good initial guess for the parameters. Due to the large running time of the neural network training, we generate 100 virtual patients randomly, and generate 1000 initial conditions and doses for each virtual patient, resulting in 100000 training data.

During therapy, we want to track changes in individual parameters, which is why we carry out sliding window estimation with neural networks. We generate the training dataset of both networks the same way, with two different experimental setup. Both setups include weekly measurements (with the exception of weekends, no measurement is done on Saturdays and Sundays), so an experimental setup includes five measurements of the total tumor volume. Setup #1 runs from Tuesday to Tuesday, and setup #2 runs from Friday to Friday. Figure 1 shows the two experimental setups. Red crosses indicate the tumor volume measurement

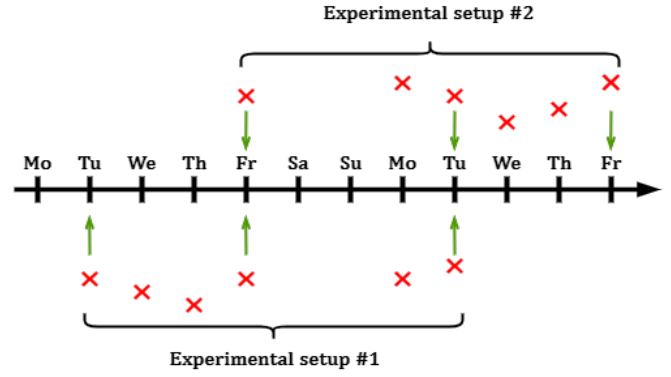


Fig. 1. The experimental setup with red crosses indicating tumor volume measurements and green arrows indicating injections.

for the current day, and green arrows indicate which days the injections occurred. During the experiment, the doses are calculated using model parameters estimated using the measurements on the previous week, thus we can use two doses for the estimation as well. As a total, both neural networks have seven inputs: the five volume measurement and two injection doses.

Thus, the inputs of the PDPK and TDNN are measurements of the total tumor volume and the injection doses, the measurements of the tumor volumes are done every day except weekends. The randomly generated 1000 initial conditions are used for all the patients. The living tumor volume at the time of the first treatment is $x_1(0)$, which is a random number between 100 mm³ and 1000 mm³, and the dead tumor volume at the same time is $x_2(0)$, which is also a random number between 10 mm³ and 200 mm³. The initial values for x_3 and x_4 are considered to be zero. For both networks we prepare each training datasets with two experimental setups.

There is noise in every dataset we use to teach and test the networks. The noise has to be added considering the properties of the measurements. The tumor volume during the experiments is estimated from caliper measurements. The measurements are done by measuring the length and the width of the tumor, and the volume is estimated using the formula

$$V = \frac{\pi}{2} (\text{width} \cdot \text{length})^{3/2}. \quad (16)$$

We add a uniform noise of $\pm 10\%$ to the length and the width measurement, and calculate the corresponding noisy tumor volumes accordingly using (16).

IV. TRAINING RESULTS

We train two four-layer neural networks, with 100 and 50 neurons in the hidden layers. We use sigmoid activation functions in the hidden layer, since they provided the best results during tests. We carried out multiple experiments with different number of neurons in the hidden layers, and also increased the number of hidden layers, and the current setup turned out to be optimal in the view of complexity and accuracy.

Parameter	a	b	n	ED_{50}	w	$x_1(0)$	$x_2(0)$
Minimum	0.17888	0.09288	0.0036	0.0062	0.1126	100	10
Maximum	0.3990	0.64812	0.20784	0.1188	0.4147	1000	200

TABLE I

THE RANGES OF THE MODEL PARAMETERS FOR THE TUMOR DYNAMICS AND THE RANGES OF THE INITIAL CONDITIONS FOR THE LIVING AND DEAD TUMOR VOLUMES.

Parameter	ED_{50}	c	k_1	k_2	$dose$
Minimum	0.0062	0.1286	70.0957	0.5746	0.2
Maximum	0.1188	12.8880	344.7991	4.9254	6

TABLE II

THE RANGES OF THE MODEL PARAMETERS FOR THE PDPK MODEL AND THE RANGE FOR THE INJECTED DOSES.

The training of the neural networks was carried out on a PC with Intel Core(TM) i7-10610U CPU @ 1.80GHz 2.30 GHz and 16GB RAM. The running time of the training of the PDPK network was 90 hours, while the running time of the training of the TDNN was 120 hours. During the training, 70% of the training data was used for actual training, 15% for validation and 15% for test. We have also generated test data with 100 virtual patients and 10 different initial conditions for each patients, thus we used 1000 data to perform tests on the neural networks.

The relative estimation errors are shown in Figures 2-5. In these figures, we show the estimation error range $\pm 50\%$, and on the top of each column we show the percentage of the dataset on which the performance was in the $\pm 50\%$ range. The red horizontal curve shows the median of the data, the blue rectangles show the range where the 25%-75% of the data are, the whiskers show data limit that are not considered as outliers, while the red crosses show the outliers.

A. Pharmacodynamic and Pharmacokinetic Neural Network

Figure 2 shows that the estimation error of the PDPK network for most of the training data is in the $\pm 50\%$ range, thus the network could learn the parameters efficiently. Note that only 70% of the training data was used for actual training, thus the network performs well on test data as well. For 50% of the data the estimation errors are in the $\pm 10\%$ range.

Figure 3 shows the estimation errors for the 1000 test data generated independently of the training data. The performance is worse than in the previous case, however, the parameters are generated from a large range, and the network was trained with a relatively small number of training data compared to the dimension of the problem. According to the figure, about 50% of the data was in the $\pm 50\%$ estimation error range, which is satisfactory for initial estimation of the parameters. Note that the pharmacokinetics generally has a component which is fast compared to the sampling time of the measurements used in the experiment, thus the problem is undersampled. Regardless of this issue, the PDPK network performed well on the training dataset and satisfactory on the test data.

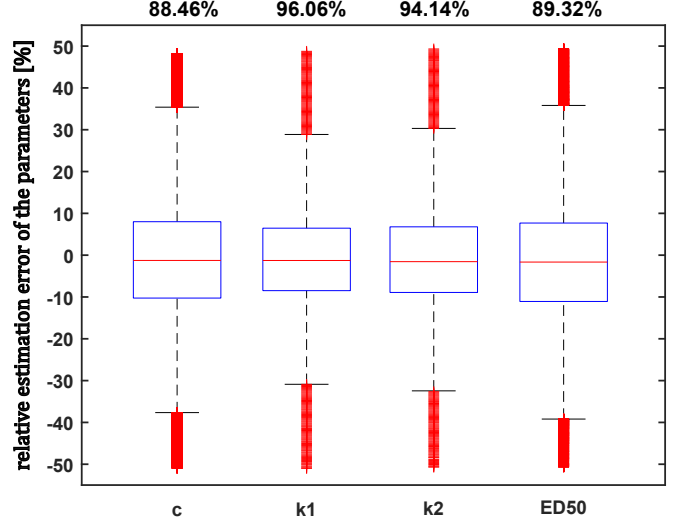


Fig. 2. The relative estimation error in the range $\pm 50\%$ on the training dataset, the percentage of data on which the PDPK network performed in the given range are over the columns.

B. Tumor Dynamics Neural Network

The performance of the TDNN on the training dataset is shown in Figure 4. The figure shows that the estimation error for most of the data was in the $\pm 50\%$ range. The lowest percentage of data in the given estimation range is for the parameters ED_{50} and n , whose sensitivity is low, as shown in [18]. The performance is better than in the case of the PDPK network, since the problem is not undersampled for the tumor dynamics model.

The performance was also tested using the 1000 test data generated independently of the training data. The results are shown in Figure 5. The results are good for the parameters a , b and w , since over 80% of the data is estimated with relative error in the $\pm 50\%$ range, but only about 50% of the test data are in the range when we consider the estimation of the parameters ED_{50} and n . This is a satisfactory result, considering the low sensitivity of these parameters. Thus, the figures show that the proliferation rate a , maximum drug effect rate b and the washout rate w can be estimated with good accuracy, thus the TDNN can be used to give a good initial guess for these three parameters. The results also prove that the estimation of the parameters ED_{50} and n are difficult in the current experimental setup.

V. CONCLUSION

We considered the problem of parameter estimation of tumor dynamics and pharmacodynamic/pharmacokinetic parameters for a given experimental setup. The estimation

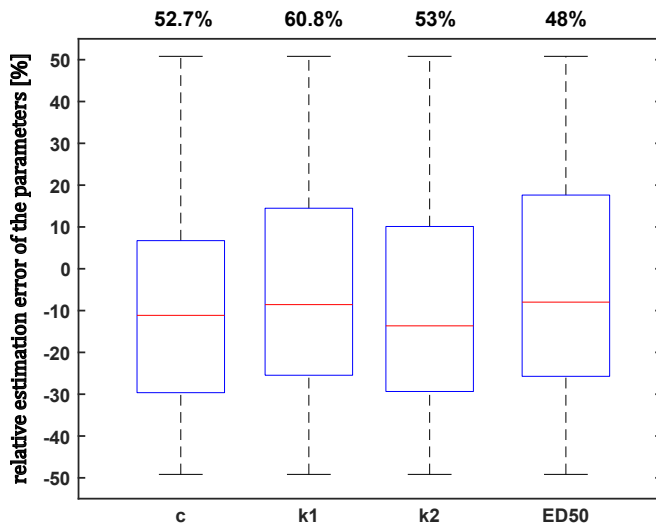


Fig. 3. The relative estimation error in the range $\pm 50\%$ on the test dataset, the percentage of data on which the PDPK network performed in the given range are over the columns.

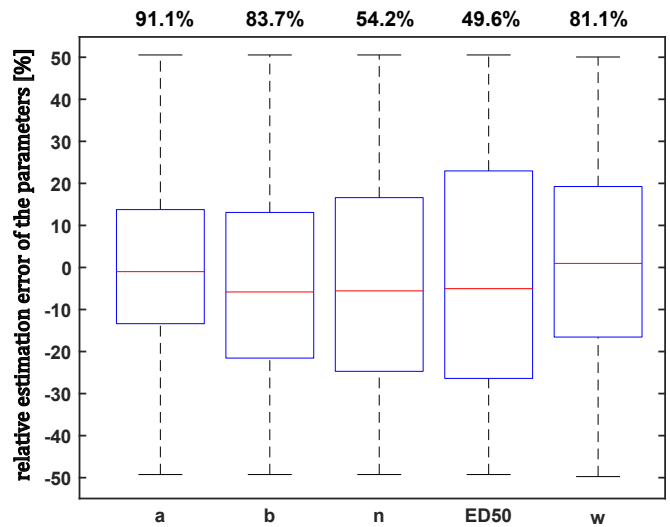


Fig. 5. The relative estimation error in the range $\pm 50\%$ on the test dataset, the percentage of data on which the TDNN performed in the given range are over the columns.

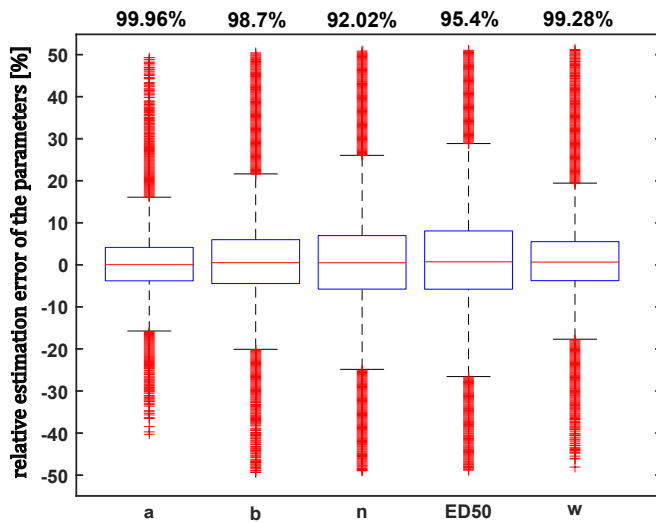


Fig. 4. The relative estimation error in the range $\pm 50\%$ on the training dataset, the percentage of data on which the TDNN performed in the given range are over the columns.

results can be used as initial estimation for local search algorithms which makes the estimation more accurate. However, finding a good initial estimation is a hard task, considering the large range in which the parameters can change. The results showed that the trained neural networks can be used to provide good initial estimations.

The current experimental setup is an undersampled problem for the identification of the pharmacokinetic parameters. Although, with the training technique used in the paper we can create neural networks that provide a good estimation for the training data and satisfactory estimation for the test data.

The median effective dose parameter (ED_{50}) and the tumor necrotic rate (n) has low sensitivity provided the measurements used in the experiments. This has been proved with sensitivity analysis in [18], and also proved here numer-

ically using the neural networks. Based on this experiment, the training technique can be used to estimate the sensitivity of the parameters for complex models.

REFERENCES

- [1] J. M. Greene, J. L. Gevertz, and E. D. Sontag, "Mathematical approach to differentiate spontaneous and induced evolution to drug resistance during cancer treatment," *JCO Clinical Cancer Informatics*, no. 3, pp. 1–20, 2019.
- [2] D. A. Drexler, T. Ferenci, A. Lovrics, and L. Kovács, "Tumor dynamics modeling based on formal reaction kinetics," *Acta Polytechnica Hungarica*, vol. 16, pp. 31–44, 2019.
- [3] A. R. Akhmetzhanov, J. W. Kim, R. Sullivan, R. A. Beckman, P. Tamayo, and C.-H. Yeang, "Modelling bistable tumour population dynamics to design effective treatment strategies," *Journal of Theoretical Biology*, vol. 474, pp. 88–102, 2019.
- [4] F. Cacace, V. Cusimano, and P. Palumbo, "Optimal impulsive control with application to antiangiogenic tumor therapy," *IEEE Transactions on Control Systems Technology*, vol. 28, no. 1, pp. 106–117, Jan 2020.
- [5] H.-P. Ren, Y. Yang, M. S. Baptista, and C. Grebogi, "Tumour chemotherapy strategy based on impulse control theory," *Philosophical Transactions Mathematical Physical & Engineering Sciences*, vol. 375, no. 2088, 2017.
- [6] D. A. Drexler and L. Kovács, "Optimization of impulsive discrete-time tumor chemotherapy," in *Proceedings of the 2019 IEEE 1st International Conference on Societal Automation*, 2019.
- [7] B. Péceli, D. A. Drexler, and L. Kovács, "Optimal scheduling of low-dose metronomic chemotherapy: an in-silico analysis," in *2020 IEEE 15th International Conference of System of Systems Engineering (SoSE)*, 2020, pp. 411–416.
- [8] J. Klamka, H. Maurer, and A. Swierniak, "Local controllability and optimal control for a model of combined anticancer therapy with control delays," *Mathematical Biosciences and Engineering*, vol. 14, no. 1, pp. 195–216, 2017.
- [9] F. S. Lobato, V. S. Machado, and V. Steffen, "Determination of an optimal control strategy for drug administration in tumor treatment using multi-objective optimization differential evolution," *Computer Methods and Programs in Biomedicine*, vol. 131, pp. 51 – 61, 2016.
- [10] D. A. Drexler, J. Sápi, and L. Kovács, " H_∞ control of nonlinear systems with positive input with application to antiangiogenic therapy," in *Proceedings of the 9th IFAC Symposium on Robust Control Design ROCOND 2018*, 2018, pp. 146 – 151.
- [11] N. A. Perestyuk, V. A. Plotnikov, A. M. Samoilenko, and N. V. Skripnik, *Differential Equations with Impulse Effects*. Berlin, Germany: De Gruyter, 2011.
- [12] T. Yang, *Impulsive Control Theory*. Berlin, Germany: Springer, 2001.

- [13] A. M. Jarrett, E. A. B. F. Lima, D. A. Hormuth, M. T. McKenna, X. Feng, D. A. Ekrut, A. C. M. Resende, A. Brock, and T. E. Yankeelov, "Mathematical models of tumor cell proliferation: A review of the literature," *Expert Review of Anticancer Therapy*, vol. 18, no. 12, pp. 1271–1286, 2018.
- [14] P. M. Altrock, L. L. Liu, and F. Michor, "The mathematics of cancer: integrating quantitative models," *Nature Reviews. Cancer*, vol. 15, no. 12, pp. 730–745, 2015.
- [15] P. Hahnfeldt, D. Panigrahy, J. Folkman, and L. Hlatky, "Tumor development under angiogenic signaling: A dynamical theory of tumor growth, treatment response, and postvascular dormancy," *Cancer Research*, vol. 59, pp. 4770–4775, 1999.
- [16] P. A. Valle, L. N. Coria, and Y. Salazar, "Tumor Clearance Analysis on a Cancer Chemo-Immunotherapy Mathematical Model," *Bulletin of Mathematical Biology*, vol. 81, no. 10, pp. 4144–4173, Oct. 2019, wOS:000487900700018.
- [17] D. A. Drexler, T. Ferenci, A. Füredi, G. Szakács, and L. Kovács, "Experimental data-driven tumor modeling for chemotherapy," in *Proceedings of the 21st IFAC World Congress*, 2020, pp. 16 466–16 471.
- [18] M. Siket, G. Eigner, and L. Kovács, "Sensitivity and identifiability analysis of a third-order tumor growth model," in *2020 IEEE 15th International Conference on System of Systems Engineering (SoSE)*, 2020, pp. 417–422.
- [19] M. Siket, G. Eigner, D. A. Drexler, I. Rudas, and L. Kovács, "State and parameter estimation of a mathematical carcinoma model under chemotherapeutic treatment," *Applied Sciences*, vol. 10, no. 24, 2020.
- [20] M. Fidler, Y. Xiong, R. Schoemaker, J. Wilkins, M. Trame, T. Post, and W. Wang, *nlmixr: Nonlinear Mixed Effects Models in Population Pharmacokinetics and Pharmacodynamics*, 2018, R package version 1.0.0-7. [Online]. Available: <https://CRAN.R-project.org/package=nlmixr>
- [21] J. Owen and J. Fiedler-Kelly, *Introduction to Population Pharmacokinetic / Pharmacodynamic Analysis with Nonlinear Mixed Effects Models*. Wiley, 2014.
- [22] M. V. Cakir, O. Eyerciogl, K. Gov, M. Sahin, and S. H. Cakir, "Comparison of soft computing techniques for modelling of the edm performance parameters," *Advances in Mechanical Engineering*, vol. 5, 2013.
- [23] D. K. Chaturvedi and M. P. Singh, "Parameter estimation and soft computing techniques," in *Proceedings of the Third International Conference on Soft Computing for Problem Solving*, M. Pant, K. Deep, A. Nagar, and J. C. Bansal, Eds. New Delhi: Springer India, 2014, pp. 267–279.
- [24] J. A. Lazzús, P. Vega-Jorquera, C. H. López-Caraballo, L. Palma-Chilla, and I. Salfate, "Parameter estimation of a generalized lotka–volterra system using a modified pso algorithm," *Applied Soft Computing*, vol. 96, p. 106606, 2020. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1568494620305445>
- [25] L. Liu, D. Ma, A. T. Azar, and Q. Zhu, "Neural computing enhanced parameter estimation for multi-input and multi-output total non-linear dynamic models," *Entropy*, vol. 22, no. 5, 2020. [Online]. Available: <https://www.mdpi.com/1099-4300/22/5/510>
- [26] M. Puskás and D. A. Drexler, "Parameter identification of a tumor model using artificial neural networks," in *Proceedings of the 2021 IEEE 19th World Symposium on Applied Machine Intelligence and Informatics*, 2021, pp. 443–448.
- [27] P. Érdi and J. Tóth, *Mathematical Models of Chemical Reactions. Theory and Applications of Deterministic and Stochastic Models*. Princeton, New Jersey: Princeton University Press, 1989.
- [28] J. Tóth, A. L. Nagy, and D. Papp, *Reaction kinetics: exercises, programs and theorems*. Springer, 2018.
- [29] H. Yu and B. Wilamowski, *Levenberg–Marquardt Training*, 02 2011.