# MODEL PREDICTIVE CONTROL OF CANCER CELLULAR DYNAMICS: A NEW STRATEGY FOR THERAPY DESIGN

Benjamin Smart[1], Irene de Cesare[2], Ludovic Renson[1], Lucia Marucci[2,3,4]

[1]Department of Mechanical Engineering, Imperial College London, London SW7 1AY, U.K.

[2]Department of Engineering Mathematics, University of Bristol, Woodland Road, Bristol BS8 1UB, U.K.

[3]BrisSynBio, Life Sciences Building, University of Bristol, Tyndall Avenue, Bristol BS8 1TQ, U.K.

[4]School of Cellular and Molecular Medicine, University of Bristol, University Walk, Bristol BS8 1TD, U.K.

e-mail: b.smart21@imperial.ac.uk, l.renson@imperial.ac.uk, lucia.marucci@bristol.ac.uk

**Abstract**

Recent advancements in Cybergenetics have led to the development of new computational and experimental platforms that enable to robustly steer cellular dynamics by applying external feedback control. Such technologies have never been applied to regulate intracellular dynamics of cancer cells. Here, we show *in silico* that adaptive model predictive control (MPC) can effectively be used to steer signalling dynamics in Non-Small Cell Lung Cancer (NSCLC) cells to resemble those of wild-type cells, and to support the design of combination therapies. Our optimisation-based control algorithm enables tailoring the cost function to force the controller to alternate different drugs and/or reduce drug exposure, minimising both drug-induced toxicity and resistance to treatment. Our results pave the way for new cybergenetics experiments in cancer cells for drug combination therapy design.

## Keywords

Adaptive Model Predictive Control (MPC), Combination Therapies, Cybergenetics, External Feedback Control, Non-Small Cell Lung Cancer (NSCLC)

# 1 Introduction

Cybergenetics is a recent field of synthetic biology, which refers to the forward engineering of complex phenotypes in living cells applying principles and techniques from control engineering [1].

Three main approaches have been proven to be effective for the control of different processes (such as gene expression and cell proliferation), namely: i) open- or closed-loop controllers embedded into cells by means of synthetic gene networks [2–6]; ii) external controllers, where the controlled processes are within cells, while the controller (either at single cell or cell-population level) and the actuation functions are implemented externally via microfluidics-optogenetics/microscopy-flow cytometry platforms and adequate algorithms for online cell output quantification and control [7–16]; iii) multicellular control, where both the control and actuation functions are embedded into cellular consortia [17–21]. Plenty of examples of embedded controllers have been engineered across different cellular chassis; instead, applications of external and multicellular controllers in mammalian cells are scarce and either just theoretical or limited to proof of concepts.

Here, we propose to apply cybergenetics, in particular external feedback control, to predict combinations of drugs (i.e. control inputs) which can bring dysregulated cellular variables (i.e. gene expression, control output of the system) within tightly controlled ranges in cancer cells. We take Non-Small Cell Lung Cancer (NSCLC) as an example; using a previously proposed differential equations mathematical model describing the dynamics of the EGFR and IGF1R pathways, we show *in silico* that external feedback controllers can effectively steer intracellular gene expression dynamics in cancer cells to resemble those of wild-type cells.

The use of feedback control is advantageous as it enables coping with changes in both steady-state levels and in the temporal dynamics of genes involved in dysregulated signalling cascades. The control action is implemented by means of adaptive model predictive control (MPC), thus not requiring an exact model of the system; this is particularly advantageous in biological applications, where the derivation of detailed models can be time-consuming and troublesome [22–24].

The possibility to control single/multiple outputs with one or more inputs can support the design of combination therapies which target different nodes in signalling cascades; this approach can be advantageous to maximise the efficacy of cancer therapies [25]. In this regard, our optimisation-based control algorithm enables tailoring the cost function to force the controller to alternate different drugs and/or reduce drug exposure. The controller should also be able to cope with the crosstalk in signalling pathways, which might be further mechanisms causing drug resistance [26].

In what follows, we demonstrate that adaptive MPC can be used to effectively steer the concentration of several proteins within different signalling pathways of a NSCLC cell in order to tune gene expression, whilst reducing the dose of each drug provided as an input. Our results pave the way for using cybergenetics for cancer treatment design.

# 2 Methods

## 2.1 Control Scheme Used in External Feedback

We applied a feedback controller to regulate the concentrations of two downstream genes (ERK and Akt) of the mTOR and MAPK pathways as modelled in [27], and as shown in Figure 1.

Figure 2 shows the response of y1(ERK) and y2(Akt) in a wild type $\sim$ and in a NSCLC $\sim$ cell to a phosphorylation of EGFR and IGF1R as modelled by varying the system's initial conditions as in [27], referred to as an activation. A wild type cells' activation is modelled using an active concentration of 8000 $\mu M$ for EGFR and 800$\mu M$ for IGF1R, while an activation in NSCLC cells is triggered by an active concentration of 800000 and 400000 $\mu M$ for EGFR and IGF1R, respectively. The term 'free' refers to an open loop response (i.e. if no feedback control

is applied) in NSCLC cells. It can be seen that y1(ERK) and y2(Akt) activation dynamics are different in cancer vs wild-type cells 2; of note, the activation of y1(ERK) occurs over a timescale of an order of magnitude faster than the activation of y2(Akt).

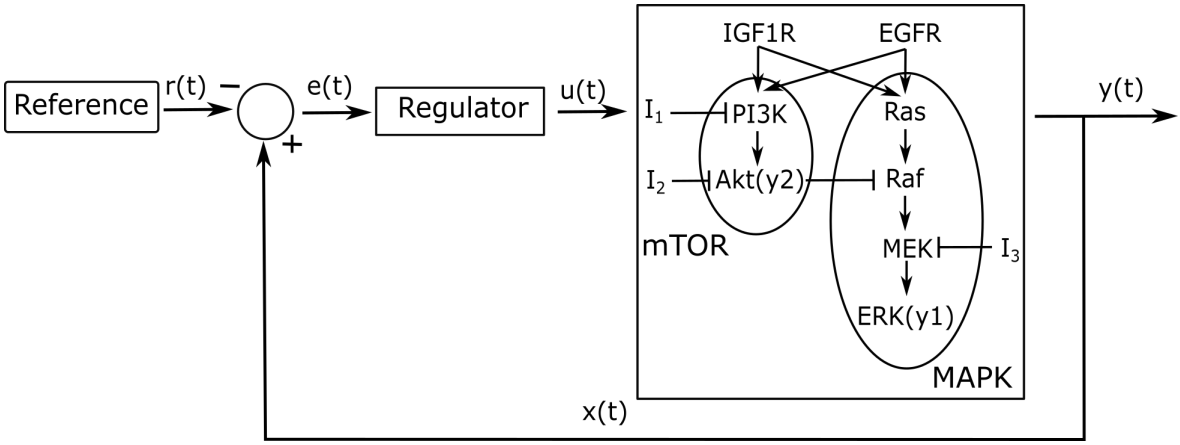


Figure 1: A control scheme including three inputs ($I_1$, $I_2$, $I_3$) that interact with the mTOR and MAPK pathways. Two observable protein concentrations, y1(ERK) and y2(Akt), are used as control outputs for the two pathways. The regulator used throughout this project is an adaptive MPC program which attempts to steer the concentrations of the outputs to the transient response of the wild type cell, set as the control reference, as shown in Figure 2.

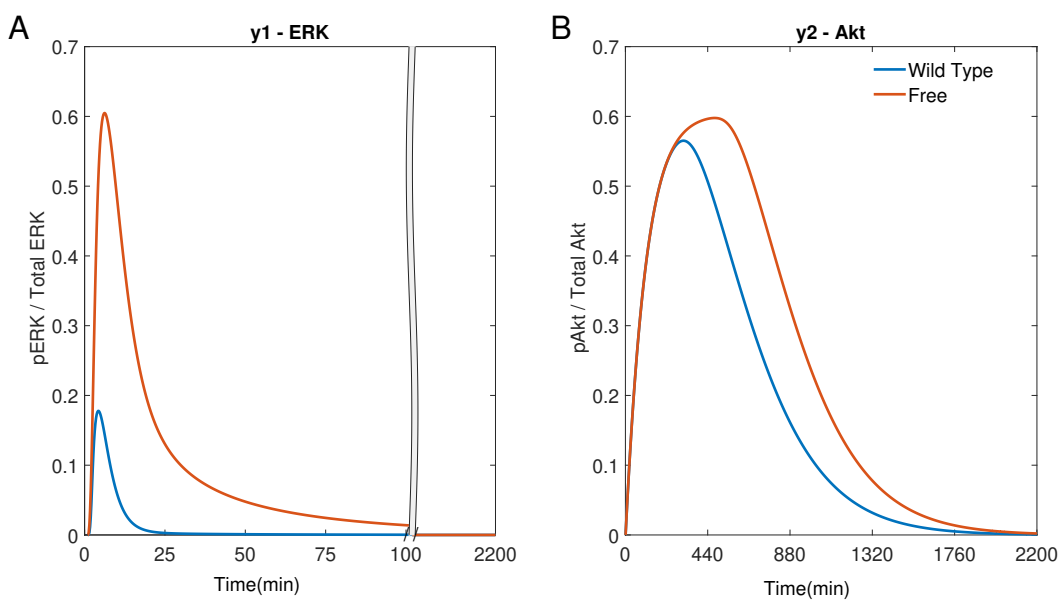


Figure 2: Simulations of the NSCLC model [27]: free NSCLC cell response ∼ compared to a wild type cell ∼ [27]. (A) The response of y1(ERK). (B) The response of y2(Akt).

The two pathways are both kinase activated cascades, meaning that an activation at the receptors in the cell membrane causes a cascade of phosphorylation in downstream genes. Therefore it is difficult to robustly control the system as, once an error is measured in the outputs, it can be too late to have a significant effect by acting on the internal states higher up the cascade.

A model based controller can interpret a small change in the outputs. On the other hand, model-free feedback strategies with large gains can have undesirable consequences, such as poor dynamic performance. A Proportional controller can be tuned to decrease the control error, but gains have to be carefully chosen, and the user cannot

impose desired constrains on the input (Section S8).

Adaptive MPC is used as the model-based control scheme here (Figure 1). The success of MPC relies on the quality of the model used to predict the future behaviour of the system, and on the cost function the controller uses to calculate the optimal inputs to be fed to the process. The novelty of the controller used here lies in the choice of adaptive model and in the cost function used. The MPC implementation is presented in Section S4.

## 2.2 MPC's Linearised Model

The NSCLC model [27] contains significant non-linear terms and a large number of internal states. An adaptive MPC controller which computes a linear approximation of the NSCLC model at each time step, predicts the future of the internal states and calculates the optimal input profile [28]. The controller then applies the first input of its calculated optimal input profile to the actual system. At the next time step, the controller calculates a new linear model by linearising the NSCLC model. The use of a linear system results in a convex optimisation problem which can be solved quickly. Adaptive MPC is used in all simulations unless stated otherwise.

Alternatively, non-linear MPC could be used; however, it is computationally expensive, and the time needed to compute the next input could be longer than the sampling/actuation temporal intervals (see Section S7).

## 2.3 Improving the Traditional MPC Cost Function

The cost function used by the adaptive MPC algorithm to find the optimal input depends on the current internal state error, $\mathbf{e}_0$, and on the inputs, $\mathbf{u}(t)$. The error, $\mathbf{e}(t)$, is the difference between the reference, $\mathbf{r}(t)$, and the internal states of the NSCLC system, $\mathbf{x}(t)$, as shown in Figure 1. The standard cost function used for linear MPC controllers [29], focuses on how readily the inputs, $\mathbf{u}(t)$, are used and on reducing the proportion error in the states, $\mathbf{e}(t)$.

In order to include both the magnitude and duration of the error, the integral of the state error, $\int \mathbf{e}(t)\,dt$, is added to the standard cost function. It has been shown that it is beneficial to integrate the state error [30], meaning that the controller acts due to these longer, smaller errors in the outputs caused by states higher in the cascade. Moreover, to avoid rapid fluctuations in the control input, $\mathbf{u}(t)$, a differential cost of the inputs is also added to cost function. The derivation of the cost function and a description of the weights of its terms $(\alpha, \beta, \gamma, \eta, \theta)$ can be found in Section S4.

## 2.4 MPC Simulation Parameters

The MPC simulations are reproducible thanks to the deterministic nature of the model and controller, as long as the cost function coefficient weights and other MPC related parameters are kept constant. Table 1 gives a summary of these parameters. Several key parameters are added to the figures' captions.

## 2.5 Indexes Used to Quantify Control Performance

To assess quantitatively the performance of our controller, we define an Error Index, $EI$. It is the sum of the squared error between the output and the reference for the outputs, as used in [31].

$$EI = \int_0^T \mathbf{e}^T \mathbf{C} \mathbf{e}\, dt \tag{1}$$

$\mathbf{C}$ is the output matrix of the linearised NSCLC model. A small $EI$ indicates a good performance of the controller.

To quantify the controller effort needed to achieve a certain output, we assess the dose of input drug(s) using a Dose Index, $DI_i$. It is the integral of the input signal, where $\mathbf{u}(t) = [I_1(t), I_2(t), I_3(t)]$.

| Parameter | Description | Value |
|---|---|---|
| $T_s$ | Sampling time of the MPC regulator | Usually 1 min (0.02 min and 30 min in Figures 6 and 10) |
| $N$ | Length of the prediction horizon | 10 steps |
| $\alpha$ | Weight of the internal state errors. | 0 |
| $\beta$ | Extra weight associated with the output state errors. | Usually 0 (1 in Figure 3A)) |
| $\gamma$ | Weight associated with each input (it is a row vector, with an element for each input). | $[0 \quad 0 \quad 0] \leq \gamma$ <br> $\gamma \leq 10^{15}[1 \quad 1 \quad 1]$ |
| $\theta$ | Weight of the gradient of the input profile. | Usually 0 ($10^5$, Figure 3C)) |
| $\eta$ | Weight of the integral of the output state errors. | Usually 1 (0 in Figure 3A)) |

Table 1: Parameters used for MPC simulations.

$$DI_i = \int_0^T I_i(t)\, dt \tag{2}$$

The inputs can never be negative as they are physical concentrations, therefore there is no need to square the input signal.

# 3 Results

Non-Small Cell Lung Cancer (NSCLC) accounts for 80% of lung cancer cases and is characterised by various mutations which usually lead to an overexpression of the EGF and IGF1R receptors. These receptors trigger several cascades including the mTOR and MAPK pathways; their downstream genes FOXO1 and C-FOS regulate cell apoptosis and proliferation. The differential equations-based mathematical model for NSCLC signalling developed in [27] includes the mTOR and MAPK pathways along with some of the reactions between the two pathways, as shown in Figure 1; the model enables comparing gene expression dynamics in wild type vs cancers cell. We chose the downstream genes ERK and Akt (noted as y1 and y2, respectively) as the control outputs for the external feedback loop (Figure 1). The two outputs can be tuned by varying three inputs ($I_1$, $I_2$, $I_3$), which inhibit three specific proteins within the mTOR and MAPK pathways. The pathways can influence each others' reactions, creating internal feedback loops (crosstalk).

The code used to implement an adaptive MPC program on this NSCLC model used in these simulations is available on GitHub: `https://github.com/Ben-Smart/Adaptive_MPC_on_NSCLC.git`.

## 3.1 Assessing the Importance of Each Term in the Cost Function

Firstly, Single-Input Single-Output (SISO) simulations were performed. The controller tries to steer the dynamics of either y1(ERK) or y2(Akt) by varying the concentrations of a drug that acts directly on one of the two signalling cascades ($I_3$ for y1(ERK) and either $I_1$ or $I_2$ for y2(Akt)). Figure 3 uses $I_2$ to regulate y2(Akt), and shows the effect of different cost function terms on the performance of the controller.

Figure 3A shows that using integral terms within the cost function reduces the error in y2(Akt), as compared to the proportional terms ($EI \rightarrow 1.945 \sim < 10.923 \sim$). However, it can be seen in Figure 3C that the controller using integral terms $\sim$ can cause fluctuations in the input. Such fluctuations are reduced when using a differential cost, which also has a lower Error Index, but higher Dose Index ($EI \rightarrow 1.690 \sim < 1.945 \sim < 10.923 \sim$, $DI_2 \rightarrow 603 \sim > 546 \sim > 129 \sim$).

The derivative term is not included in further simulations; the cost functions used in the following simulations include $\beta = 0$, $\theta = 0$ and $\eta = 1$, as in Figure 3C, apart from the weight associated with each input, $\gamma$, that is varied (as indicated in figures' captions).

### 3.1.1 Single-Input Single-Output Control with the Chosen Cost Function

Figure 4 shows SISO adaptive MPC simulations for $I_1$, $I_2$ and $I_3$; each drug is used to control the downstream molecule in the cascade it acts on. It can be seen that plots C) and D) of Figure 4 are identical to $\sim$ in plot A) and plot C) of Figure 3 as these are both SISO responses of $I_2$ using the chosen cost function (costing the integral and input terms). All of the SISO controllers move the NSCLC response towards the wild type response $\sim$ using a lower dose than just a step of each input at the maximum allowed dose ($1\mu M$), decreasing the Dose Indexes. The step input response can be found in Section S3. This demonstrates the benefits of using an external feedback loop compared to an open loop response with a static step input.

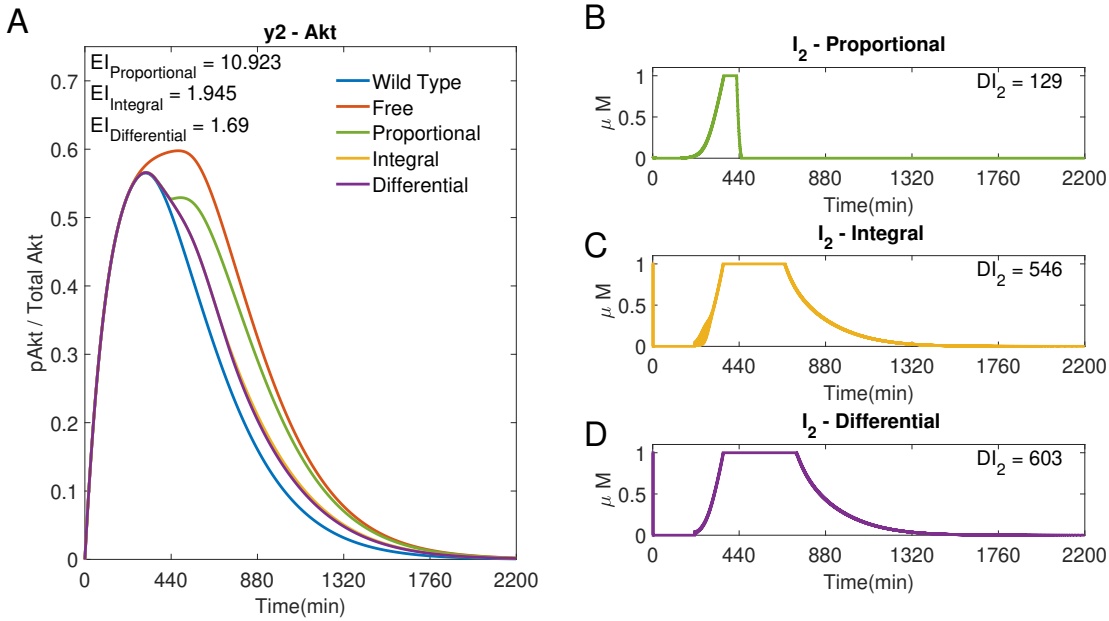


Figure 3: SISO adaptive MPC simulation using $I_2$ to control y2 (Akt), and comparing different cost functions. (A) The response of y2 to different cost functions. (B) The input profile ($I_2$) using the proportional error $\sim$ within the cost function. $\beta = 1$, $\gamma = [-, 10^5, -]$, $\theta = 0$ and $\eta = 0$. (C) The input profile ($I_2$) using the integral error $\sim$. $\beta = 0$, $\gamma = [-, 10^5, -]$, $\theta = 0$ and $\eta = 1$. (D) The input profile ($I_2$) using the integral error and input differential terms $\sim$ of the cost function. $\beta = 0$, $\gamma = [-, 10^5, -]$, $\theta = 10^5$ and $\eta = 1$. $T_s = 1min$ and $N = 10$ for all plots.

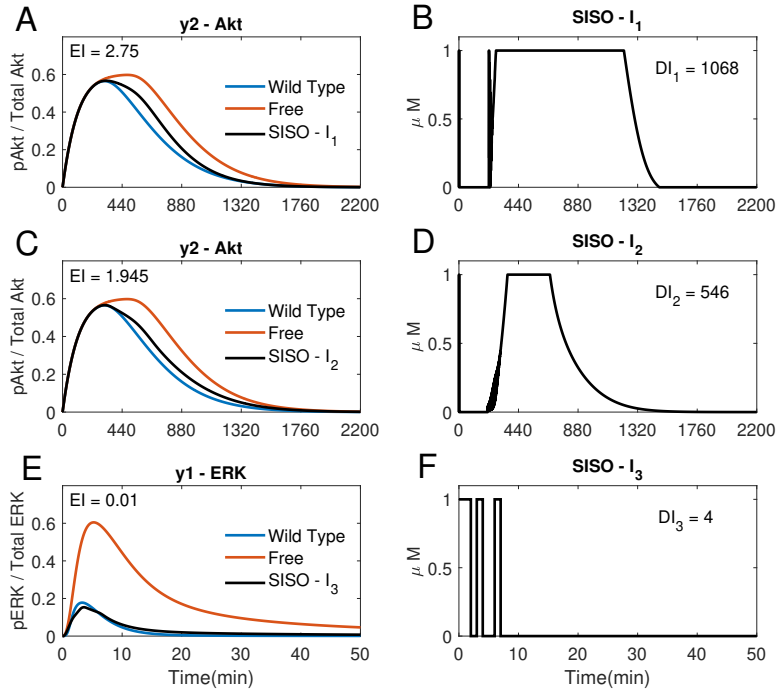


Figure 4: SISO adaptive MPC simulations. (A, B) The response of y2(Akt, in A) to the input of $I_1$ (shown in B); $\gamma = [1, -, -]$. (C, D) The response of y2 (Akt, in C) to the input of $I_2$ (shown in D); $\gamma = [-, 10^5, -]$. (E, F) The response of y1(ERK, in E) to the input of $I_3$ (shown in F); $\gamma = [-, -, 10^9]$. (A-F): $T_s = 1$, $N = 10$, $\alpha = 0$, $\beta = 0$, $\theta = 0$ and $\eta = 1$.

## 3.2 Multi-Input Multi-Output Control

Adaptive MPC can also be used to steer both outputs using all three inputs in Multi-Input Multi-Output (MIMO) simulations, as shown in Figure 5.

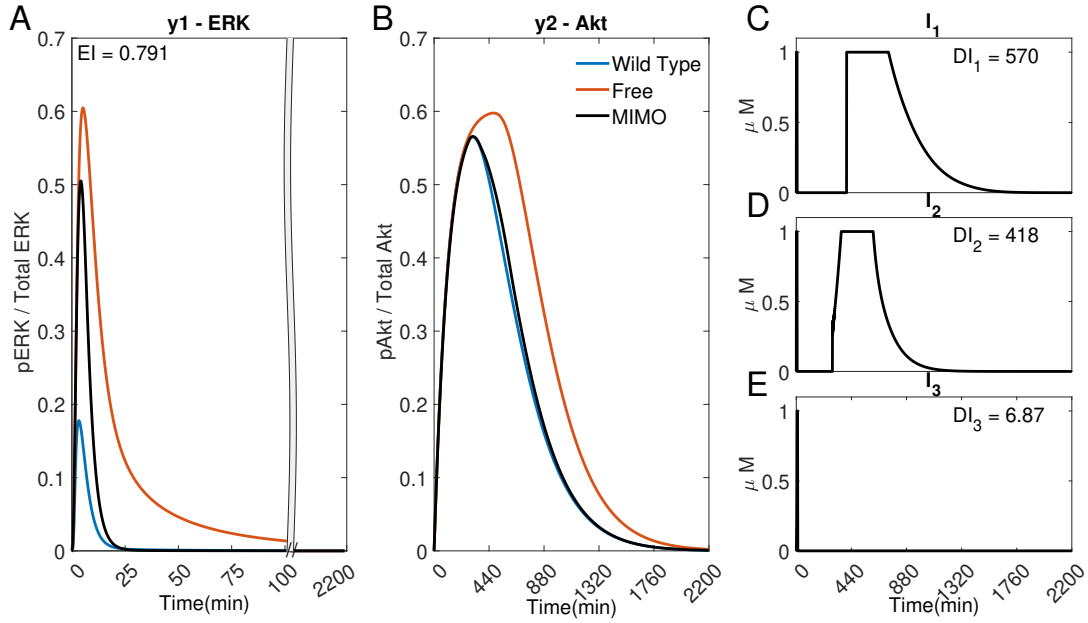


Figure 5: A MIMO adaptive MPC simulation using $I_1$, $I_2$ and $I_3$ to control the concentrations of y1 (ERK) and y2 (Akt). (A) The response of ERK. (B) The response of Akt. (C-E) The inputs used in the simulation. Parameters: $T_s = 1min$, $N = 10$, $\alpha = 0$, $\beta = 0$, $\gamma = [1, 10^5, 10^9]$, $\theta = 0$ and $\eta = 1$.

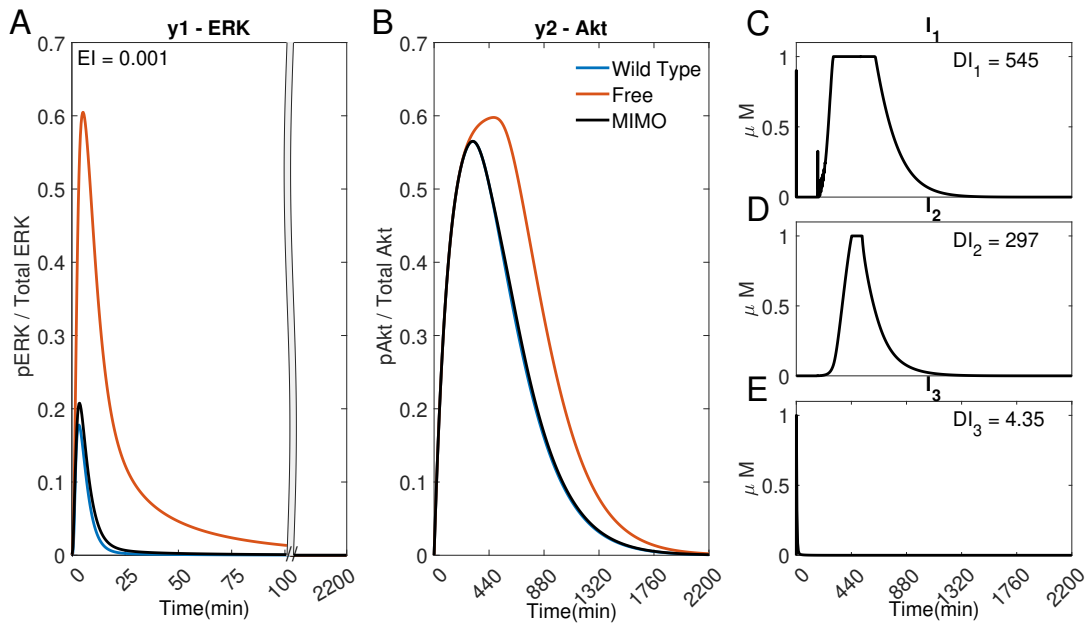


Figure 6: A MIMO adaptive MPC simulation using $I_1$, $I_2$ and $I_3$ to control the concentrations of y1 (ERK) and y2 (Akt). (A) The response of ERK. (B) The response of Akt. (C-E) The inputs used in the simulation. Parameters: $T_s = 0.02min$, $N = 10$, $\alpha = 0$, $\beta = 0$, $\gamma = [1, 10^5, 10^9]$, $\theta = 0$ and $\eta = 1$.

The error of y2 (Akt) (Figure 5B)) is significantly smaller in comparison to Figure 4A and C ($EI \rightarrow 0.791 <$

$1.954 < 2.75$), whilst using significantly less $I_1$ ($DI_1 \rightarrow 570 < 1068$) and $I_2$ ($DI_2 \rightarrow 418 < 546$), suggesting that it might be advantageous to use adaptive MPC to predict and apply combination drug profiles.

However, due to the fast dynamics of the MAPK pathway, the output y1(ERK) fails to adequately follow the reference activation curve. Figure 6 shows that if the time step is adequately reduced (for instance, to $T_s = 0.02$ minutes), the controller can handle the faster dynamics of the pathway and effectively control both outputs ($EI \rightarrow 0.001 < 0.791$) whilst using a lower dosage of all the inputs ($DI_1 \rightarrow 545 < 570$, $DI_2 \rightarrow 297 < 418$, $DI_3 \rightarrow 4.35 < 6.87$). Figure 6 shows that the controller can overcome the effect of crosstalk, balancing the error in y1(ERk) and the use of $I_1$ and $I_2$.

To emulate capabilities of current microfluidics devices, the time step will be kept at $T_s = 1min$. Therefore, in what follows, we investigate only Multi-Input Single-Output (MISO) simulations where y2 (Akt) is controlled only with $I_1$ and $I_2$. In this way, we remove the effect of the crosstalk (through the inherent negative feedback loops shown in Figure 1), which would restrict the MPC's use of $I_1$ and $I_2$ caused by the high error in y1(ERK) due to the faster timescale of this output at the controller's current time step ($T_s = 1min$).

### 3.3 Combination Therapies Using $I_1$ and $I_2$

If cells are exposed to drugs for an extended period of time, side effects and resistance might become an issue [25]. The controller could be used to find potential drug combinations that can achieve a low Error Index ($EI$) whilst reducing the dose of the inputs ($DI_i$). The weight associated with using each input, $\gamma$, within the cost function can be varied for this aim, as shown in Figure 7. The Bliss Independence (BI) formula [32, 33] has been used here as a normalised Dose Index to summarise the combined effect of multiple drugs (see S5).

Figure 7 focuses on the control of y2 (Akt) using $I_1$ and $I_2$ as control inputs (MISO control). The weights in the cost function associated with each input can be varied as a ratio of $R = \frac{\gamma_2}{\gamma_1}$, ranging from low $R$ values (where a high weight is associated with $I_1$, ($\gamma_1$), thus producing a SISO-like simulation only using $I_2$), all the way through to a high $R$ value (where $\gamma_2$ is relatively large and the controller will only use $I_1$). Figure 7 compares the normalised Error Index, $\hat{EI}$ $\sim$, and the Bliss Independence $BI$ $\sim$ to the weight ratio ($R$). It shows that there is a range of $R$ which can significantly reduce both $EI$ $\sim$ and $BI$ $\sim$. Therefore, the control performance of the MISO controller is better than any SISO simulation while keeping drug concentrations low. For the purpose of designing combination therapies, here the optimal input is associated to the minimum value of the $EI$ $\sim$.

It can be seen from Figure 7 that the minimum occurs when $R = 10^5$, corresponding to $\gamma_1 = 1$ and $\gamma_2 = 10^5$. Figure 8 compares the responses obtained using a very low or high $R$ value to the MISO simulation at the optimum of $EI$. This optimum achieves a significantly lower Error Index ($EI \rightarrow 0.25 \sim < 1.95 \sim < 2.75 \sim$), and a lower Dose Index ($DI_1 \rightarrow 568 \sim < 1068 \sim$ and $DI_2 \rightarrow 423 \sim < 546 \sim$).

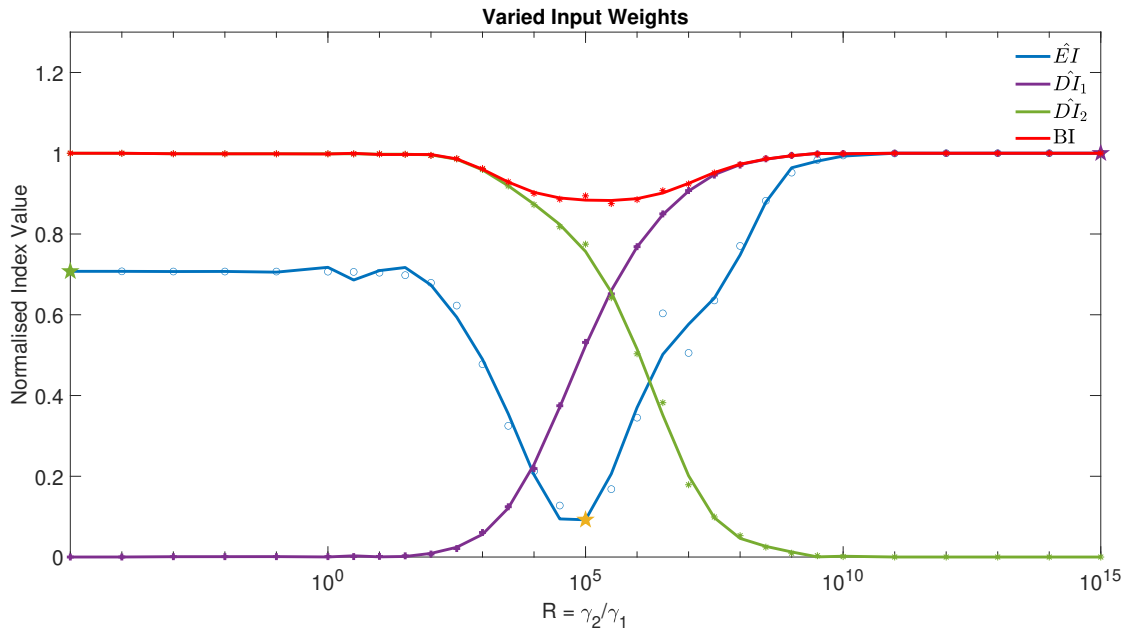


Figure 7: The normalised Error Index of y2 (Akt) ∼, Dose Indexes of $I_1$ ∼ and $I_2$ ∼ and Bliss Independence ∼, from 31 MISO adaptive MPC simulations using varied ratio of input weights, $R = \gamma_2/\gamma_1$, for example, when $R = 100$, $\gamma = [1, 100, -]$. Parameters: $T_s = 1min$, $N = 10$, $\alpha = 0$, $\beta = 0$, $\theta = 0$ and $\eta = 1$. The three star markers show the ratio used in the plots of Figure 8

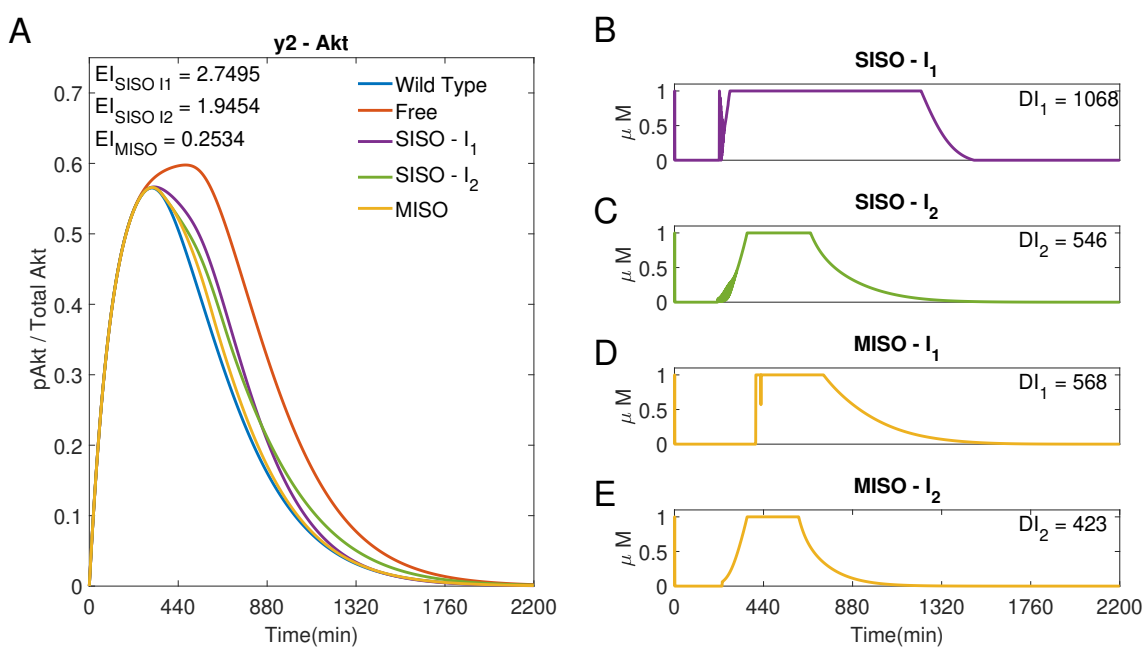


Figure 8: Three adaptive MPC simulations using $I_1$ and $I_2$ to control y2(Akt). (A) The response of y2 (Akt) to three different input weightings, SISO - $I_1$ ∼ ($\gamma = [1, 10^{15}, -]$), SISO - $I_2$ ∼ ($\gamma = [1, 10^{-5}, -]$) and MISO ∼ ($\gamma = [1, 10^{5}, -]$). $\gamma$ was selected from the minimum point of the $EI$ and the limits of $R$ in Figure 7. (B, C) Inputs $I_1$ and $I_2$ used in the two SISO simulations. (D, E) Inputs $I_1$ and $I_2$, respectively, used in the MISO simulation. Parameters: $T_s = 1min$, $N = 10$, $\alpha = 0$, $\beta = 0$, $\theta = 0$ and $\eta = 1$

### 3.4 Drug Holidays

If using an adaptive MPC, the user can set specific time intervals in which the controller does not give specific drugs (for example, to avoid toxicity induced by long exposure). These Drug Holidays can be achieved by the controller by varying the weights associated with each input, online, during a single simulation.

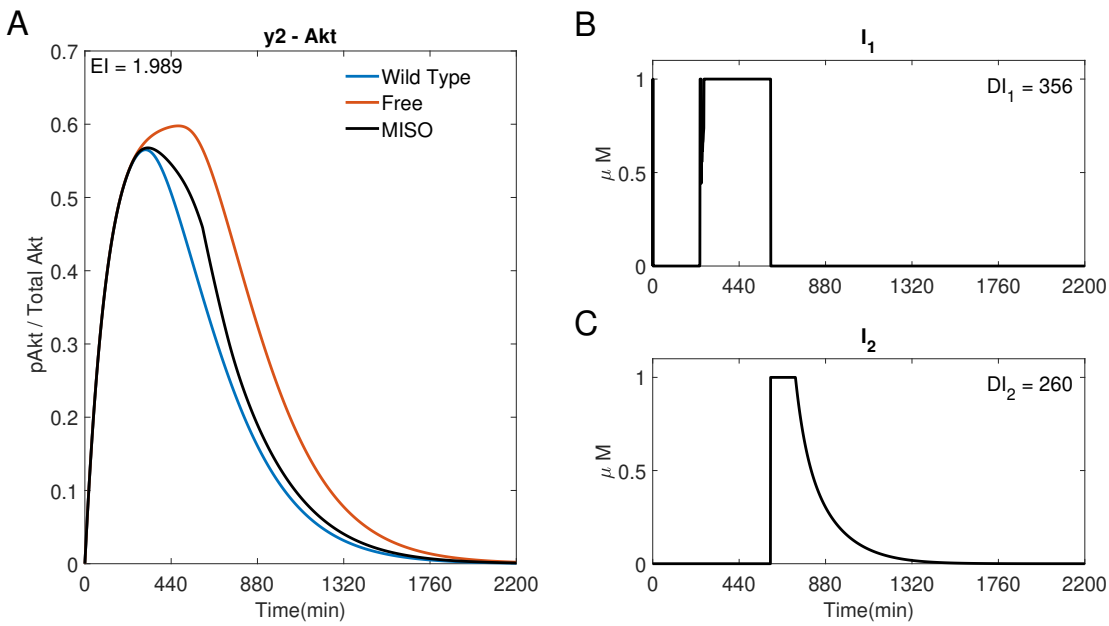


Figure 9: A MISO adaptive MPC simulation swapping $I_1$ and $I_2$ to control y2 (Akt). (A) The response of Akt. (B, C) The inputs used in the simulation. Parameters: $T_s = 1min$, $N = 10$, $\alpha = 0$, $\beta = 0$, $\gamma = [1, \infty, -]$ when $0 \leq t \leq 600min$ and $\gamma = [\infty, 10^5, -]$ when $600 < t \leq 2200min$, $\theta = 0$ and $\eta = 1$.

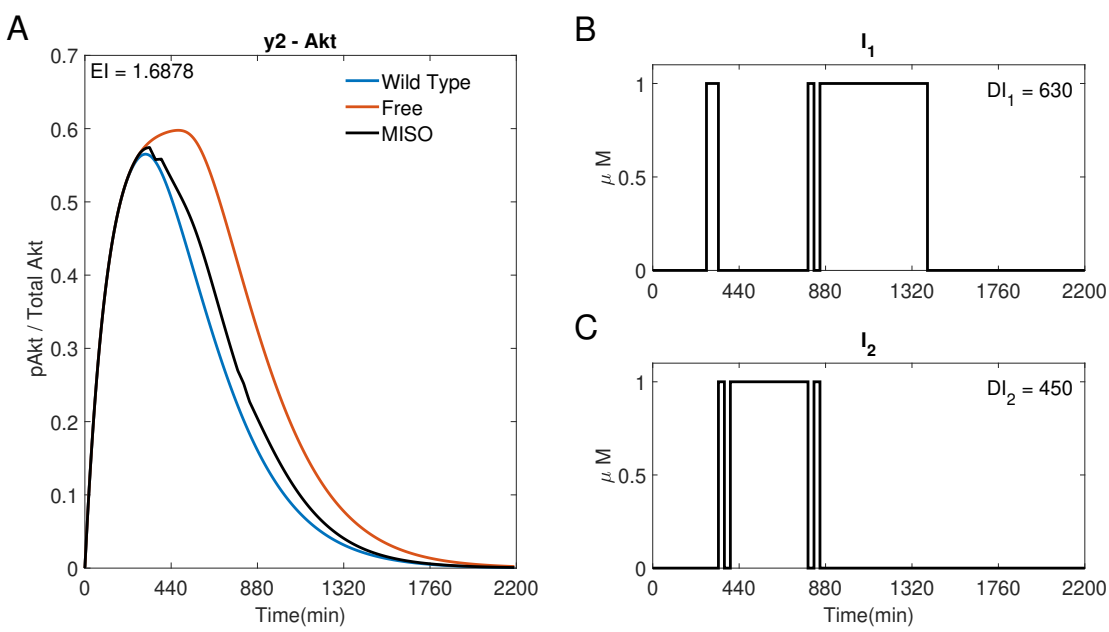


Figure 10: A discrete MISO adaptive MPC simulation using $I_1$ and $I_2$ to control y2 (Akt). (A) The response of Akt to the inputs in (B) and (C). Parameters: $T_s = 30min$, $N = 10$, $\alpha = 0$, $\beta = 0$, $\gamma = [1, 10^5, -]$, $\theta = 0$ and $\eta = 1$.

As an example, Figure 9 shows that the controller can retain a low Error Index whilst swapping inputs after 600

minutes ($EI \rightarrow 1.989 \approx 1.950 < 2.75$, the SISO $EI$ of Figure 4). Therefore it is shown that a programmed change of cost function weights during the simulation can decide which input to stop using.

Alternatively, the controller can be set to only choose one input at each time step. The inputs shown in Figure 10 have an ON or OFF state, $1\mu M$ or $0\mu M$ (discrete inputs). The step size, $T_s$, used makes a significant difference to the response. When $T_s = 1min$ the drugs can switch ON or OFF every minute, leading to rapidly fluctuating inputs (shown in Section S6). Figure 10 shows the discrete simulation with a larger time step ($T_s = 30min$). It can be seen that there is a better performance when compared to the SISO simulations (Figure 4) as $EI \rightarrow 1.6878 < 1.95 < 2.75$, whilst still using less of each input ($DI_1 \rightarrow 630 < 1068$, $DI_2 \rightarrow 450 < 546$). However, when compared to the optimal MISO response (Figure 8), these added constraints result in a higher Error Index ($EI \rightarrow 1.6878 > 0.2534$) and a higher dose ($DI_1 \rightarrow 630 > 568$, $DI_2 \rightarrow 450 > 423$).

# 4 Discussion

Computational methods have been extensively used in the search for effective cancer treatments, with approaches including optimal control to regulate dynamics of different cell populations [34–43], and a feedback action to account for changes in the cancer system [44–51]. This is, to the best of our knowledge, the first attempt of using feedback control to regulate intracellular dynamics in cancer cells.

We showed that an adaptive MPC program can be used to inform treatments for NSCLC cells, steering the dynamics of several key signalling pathways, whilst offering a tunable cost function that allows to adjust the characteristics of an optimal input. Indeed, the controller can be tuned to choose different drug profiles that will achieve a similar control performance whilst reducing exposure to one or more drugs.

Other control strategies, like PID controllers, cannot be directly tuned to effect the desired input and only act on the observed output. The use of a linear model within the MPC algorithm makes the control algorithm running time short enough for it to be used, in the future, in external feedback control experiments. The implementation of those would require some practical aspects to be considered, which we did not account for. Firstly, there might be delays in cell responses to drugs/actuation, which the model used by the controller should account for. Also, the sampling/actuation time might need to be fast enough, if aiming at controlling genes with fast dynamics. This issue might be overcome using experimental optogenetics-based platforms instead of microfluidics-based ones, as they can reduce delays in the actuation. Finally, this method assumes the knowledge of a model. In the future we hope to work on an controller that adapts the model based on online system measurements.

We foresee a growing interest in applying cybergenetics approaches, and in particular feedback controllers, to steer mammalian cells dynamics. If we realise our ambition to implement the experiments proposed here on living cells and, longer term, on patient-derived organoids, feedback control might be a valuable tool for the design or personalised optimal treatments for a range of conditions.

# 5 Conclusions

It has been demonstrated that adaptive MPC can be used to inform treatments for NSCLC cells, guiding the behaviour of two key signalling pathways and offering a tunable cost function to modify *ad hoc* the rapidity of inputs' changes, and the ways inputs as chose. The weights can also be changed during its operation to choose different drug profiles that will achieve a similar control performance whilst giving the cell a break from individual drugs. In the future, we hope to test the controller in living cells using microfluidics/microscopy platforms.

# 6 Author Information

## 6.1 Author Contributions

B.S. designed and carried out the simulations in Matlab2021b and wrote the manuscript with inputs from all the authors. I.d.C. supported the project. L.R. and L.M. conceived the project, contributed to manuscript writing and supervised the entire work.

## 6.2 Declaration of Interests

Nothing to declare.

# 7 Acknowledgements

B.S. is supported by an EPSRC DTP Scholarship. L.R. is funded by a Research Fellowship from the Royal Academy of Engineering (RF1516/15/11). L.M. is supported by the Engineering and Physical Sciences Research Council (EPSRC) grant EP/S01876X/1, and by the European Union's Horizon 2020 under Grant Agreement No. 766840.

# References


[1] M. Khammash et al. Cybergenetics: Theory and Methods for Genetic Control System. *Proceedings of the IEEE Conference on Decision and Control*, 2019-December:916–926, 12 2019. doi:10.1109/CDC40024.2019.9030209.

[2] E. Pedone et al. A tunable dual-input system for on-demand dynamic gene expression regulation. *Nature communications*, 2019. doi:10.1038/s41467-019-12329-9.

[3] C. Ciar´ et al. Synthetic negative feedback circuits using engineered small RNAs. *Nucleic Acids Research*, 46(18):9875–9889, 2018. doi:10.1093/nar/gky828.

[4] S. Aoki et al. A universal biomolecular integral feedback controller for robust perfect adaptation. *Nature 2019 570:7762*, 570(7762):533–537, 6 2019. doi:10.1038/s41586-019-1321-1.

[5] V. Hsiao et al. Design and implementation of a biomolecular concentration tracker. *ACS Synthetic Biology*, 4(2):150–161, 2 2015. doi:10.1021/SB500024B.

[6] R. Bloom et al. Synthetic feedback control using an RNAi-based gene-regulatory device. *Journal of Biological Engineering*, 9(1):1–13, 4 2015. doi:10.1186/S13036-015-0002-3.

[7] B. Shannon et al. In Vivo Feedback Control of an Antithetic Molecular-Titration Motif in Escherichia coli Using Microfluidics. *ACS Synthetic Biology*, 9(10):2617–2624, 10 2020. doi:10.1021/ACSSYNBIO.0C00105.

[8] F. Menolascina et al. In-Vivo Real-Time Control of Protein Expression from Endogenous and Synthetic Gene Networks. *PLoS Comput Biol*, 10(5):1003625, 2014. doi:10.1371/journal.pcbi.1003625.

[9] A. Milias-Argeitis et al. In silico feedback for in vivo regulation of a gene expression circuit. *Nature Biotechnology 2011 29:12*, 29(12):1114–1116, 11 2011. doi:10.1038/nbt.2018.

[10] L. Postiglione et al. Regulation of Gene Expression and Signaling Pathway Activity in Mammalian Cells by Automated Microfluidics Feedback Control. *ACS Synthetic Biology*, 7(11):2558–2565, 11 2018. doi:10.1021/ACSSYNBIO.8B00235.

[11] J. Toettcher et al. Light-based feedback for controlling intracellular signaling dynamics. *Nature methods*, 8(10):837, 10 2011. doi:10.1038/NMETH.1700.

[12] J. Uhlendorf et al. Long-term model predictive control of gene expression at the population and single-cell levels. *SYSTEMS BIOLOGY*, 109(35):14271–14276, 2012. doi:10.1073/pnas.1206810109.

[13] I. De Cesare et al. ChipSeg: An Automatic Tool to Segment Bacterial and Mammalian Cells Cultured in Microfluidic Devices. *ACS Omega*, 6(4):2473–2476, 2 2021. doi:10.1021/ACSOMEGA.0C03906.

[14] E. Pedone et al. Cheetah: A Computational Toolkit for Cybergenetic Control. *ACS Synthetic Biology*, 10(5):979–989, 5 2021. doi:10.1021/ACSSYNBIO.0C00463.

[15] M. Khazim et al. A Microfluidic/Microscopy-Based Platform for on-Chip Controlled Gene Expression in Mammalian Cells. *Methods in molecular biology (Clifton, N.J.)*, 2229:205–219, 2021. doi:10.1007/978-1-0716-1032-9{_}10.

[16] M. Khazim et al. Towards automated control of embryonic stem cell pluripotency. *IFAC-PapersOnLine*, 52(26):82–87, 1 2019. doi:10.1016/J.IFACOL.2019.12.240.

[17] N. Kylilis et al. Tools for engineering coordinated system behaviour in synthetic microbial consortia. *Nature Communications 2018 9:1*, 9(1):1–9, 7 2018. doi:10.1038/s41467-018-05046-2.

[18] G. Fiore et al. Design of a multicellular feedback control strategy in a synthetic bacterial consortium. *2016 IEEE 55th Conference on Decision and Control, CDC 2016*, pages 3338–3343, 12 2016. doi:10.1109/CDC.2016.7798771.

[19] G. Fiore et al. In-Silico Analysis and Implementation of a Multicellular Feedback Control Strategy in a Synthetic Bacterial Consortium. *ACS Synthetic Biology*, 6(3):507–517, 3 2017. doi:10.1021/ACSSYNBIO.6B00220.

[20] L. Postiglione et al. A strategy for multicellular feedback control in mammalian cells. *2019 18th European Control Conference, ECC 2019*, pages 2669–2674, 6 2019. doi:10.23919/ECC.2019.8796200.

[21] A. Matyjaszkiewicz et al. BSim 2.0: An Advanced Agent-Based Cell Simulator. *ACS Synthetic Biology*, 6(10):1969–1972, 10 2017. doi:10.1021/ACSSYNBIO.7B00121.

[22] L. Marucci. Nanog Dynamics in Mouse Embryonic Stem Cells: Results from Systems Biology Approaches. *Stem cells international*, 2017, 2017. doi:10.1155/2017/7160419.

[23] L. Marucci et al. Derivation, identification and validation of a computational model of a novel synthetic regulatory network in yeast. *Journal of mathematical biology*, 62(5):685–706, 5 2011. doi:10.1007/S00285-010-0350-Z.

[24] A. Browning et al. Identifiability analysis for stochastic differential equation models in systems biology. *Journal of the Royal Society Interface*, 17(173), 12 2020. doi:10.1098/RSIF.2020.0652.

[25] R. Salgia et al. The genetic/non-genetic duality of drug 'resistance' in cancer. *Trends in cancer*, 4(2):110, 2 2018. doi:10.1016/J.TRECAN.2018.01.001.

[26] N. Vasan et al. A view on drug resistance in cancer. *Nature 2019 575:7782*, 575(7782):299–309, 11 2019. doi:10.1038/s41586-019-1730-1.

[27] F. Bianconi et al. Computational model of EGFR and IGF1R pathways in lung cancer: A Systems Biology approach for Translational Oncology. *Biotechnology Advances*, 30(1):142–153, 1 2012. doi:10.1016/J.BIOTECHADV.2011.05.010.

[28] P. Mendis et al. Adaptive Model Predictive Control with Successive Linearization for Distillate Composition Control in Batch Distillation. *MERCon 2019 - Proceedings, 5th International Multidisciplinary Moratuwa Engineering Research Conference*, pages 366–369, 7 2019. doi:10.1109/MERCON.2019.8818777.

[29] J. Rawlings et al. *Model predictive control : theory, computation, and design.* Nob Hill Publishing, 2nd edition, 2020.

[30] J. Hornberg et al. Control of MAPK signalling: from complexity to what really matters. *Oncogene*, 24(36):5533–5542, 8 2005. doi:10.1038/SJ.ONC.1208817.

[31] G. Fiore et al. In Vivo Real-Time Control of Gene Expression: A Comparative Analysis of Feedback Control Strategies in Yeast. *ACS Synthetic Biology*, 5(2):154–162, 2 2015. doi:10.1021/ACSSYNBIO.5B00135.

[32] V. Vakil et al. Drug Combinations: Mathematical Modeling and Networking Methods. *Pharmaceutics*, 11(5), 5 2019. doi:10.3390/PHARMACEUTICS11050208.

[33] E. Demidenkoid et al. Statistical determination of synergy based on Bliss definition of drugs independence. *PLOS ONE*, 2019. doi:10.1371/journal.pone.0224137.

[34] T. Cova et al. Computational Approaches in Theranostics: Mining and Predicting Cancer Data. *Pharmaceutics 2019, Vol. 11, Page 119*, 11(3):119, 3 2019. doi:10.3390/PHARMACEUTICS11030119.

[35] A. Konstorum et al. Addressing current challenges in cancer immunotherapy with mathematical and computational modelling. *J. R. Soc. Interface*, 14, 2017. doi:10.1098/rsif.2017.0150.

[36] Khalid Kassara. A set-valued approach to control immunotherapy. *Mathematical and Computer Modelling*, 44(11-12):1114–1125, 12 2006. doi:10.1016/J.MCM.2006.03.016.

[37] F. Castiglione et al. Optimal Control in a Model of Dendritic Cell Transfection Cancer Immunotherapy. *Bulletin of Mathematical Biology*, 68:255–274, 2006. doi:10.1007/s11538-005-9014-3.

[38] A. Ghaffari et al. Optimal therapeutic protocols in cancer immunotherapy. *Computers in Biology and Medicine*, 40(3):261–270, 3 2010. doi:10.1016/J.COMPBIOMED.2009.12.001.

[39] B. Piccoli et al. Optimal vaccine scheduling in cancer immunotherapy. *Physica A*, 370:672–680, 2006. doi:10.1016/j.physa.2006.03.011.

[40] F. Castiglione et al. Cancer immunotherapy, mathematical modeling and optimal control. *Journal of Theoretical Biology*, 247:723–732, 2007. doi:10.1016/j.jtbi.2007.04.003.

[41] M. Itik et al. Optimal control of drug therapy in cancer treatment. *Nonlinear Analysis*, 71:1473–1486, 2009. doi:10.1016/j.na.2009.01.214.

[42] U. Ledzewicz et al. Antiangiogenic therapy in cancer treatment as an optimal control problem. *SIAM Journal on Control and Optimization*, 46(3):1052–1079, 2007. doi:10.1137/060665294.

[43] S. Oke et al. Optimal Control Analysis of a Mathematical Model for Breast Cancer. *Mathematical and Computational Applications 2018, Vol. 23, Page 21*, 23(2):21, 4 2018. doi:10.3390/MCA23020021.

[44] A. Goharrizi et al. Development of a new control strategy for 3D MRI-controlled interstitial ultrasound cancer therapy. *Medical Physics*, 40(3):033301, 3 2013. doi:10.1118/1.4793261.

[45] S. Chareyron et al. Model-free feedback design for mixed cancer therapy. *Biotechnology Progress*, 25(3):690–700, 5 2009. doi:10.1002/BTPR.114.

[46] A. Bratus et al. Solution of the Feedback Control Problem in the Mathematical Model of Leukaemia Therapy. *J Optim Theory Appl*, 159:590–605, 2013. doi:10.1007/s10957-013-0324-6.

[47] B. Tang et al. A Feedback Control Model of Comprehensive Therapy for Treating Immunogenic Tumours. *International Journal of Bifurcation and Chaos*, 26(3):1650039, 2016. doi:10.1142/S0218127416500395.

[48] U. Ledzewicz et al. Optimal Combined Radio- and Anti-Angiogenic Cancer Therapy. *Journal of Optimization Theory and Applications*, 180:321–340, 2019. doi:10.1007/s10957-018-1426-y.

[49] M. Alamir et al. Robust feedback design for combined therapy of cancer. *Control Appl. Meth*, 35:77–88, 2014. doi:10.1002/oca.2057.

[50] F. Teles et al. Cancer therapy optimization based on multiple model adaptive control. *Biomedical Signal Processing and Control*, 48:255–264, 2019. doi:10.1016/j.bspc.2018.09.016.

[51] J. Sápi et al. Linear state-feedback control synthesis of tumor growth control in antiangiogenic therapy. *IEEE 10th Jubilee International Symposium on Applied Machine Intelligence and Informatics, SAMI 2012 - Proceedings*, pages 143–148, 2012. doi:10.1109/SAMI.2012.6208945.

[52] Y. Wu et al. Dual role of 3-methyladenine in modulation of autophagy via different temporal patterns of inhibition on class I and III phosphoinositide 3-kinase. *The Journal of biological chemistry*, 285(14):10850–10861, 4 2010. doi:10.1074/JBC.M109.080796.

[53] H. Yang et al. Oridonin sensitizes cisplatin-induced apoptosis via AMPK/Akt/mTOR-dependent autophagosome accumulation in A549 cells. *Frontiers in Oncology*, 9(AUG):769, 2019. doi:10.3389/FONC.2019.00769.

[54] E. Martinelli et al. Antitumor activity of pimasertib, a selective MEK 1/2 inhibitor, in combination with PI3K/mTOR inhibitors or with multi-targeted kinase inhibitors in pimasertib-resistant human lung and colorectal cancer cells. *International journal of cancer*, 133(9):2089–2101, 11 2013. doi:10.1002/IJC.28236.

[55] PI3K Inhibitor Review. URL: https://www.selleckchem.com/PI3K.html.

[56] Akt Inhibitor Review. URL: https://www.selleckchem.com/Akt.html.

[57] MEK Inhibitor Review. URL: https://www.selleckchem.com/MEK.html.

[58] B. Ogunnaike et al. *Process dynamics, modeling, and control*. Oxford University Press, New York and NY, 1994.

[59] I. Bonis et al. A linear model predictive control algorithm for nonlinear large-scale distributed parameter systems. *AIChE Journal*, 58(3):801–811, 3 2012. doi:10.1002/AIC.12626.

[60] A. Albalasie et al. Using Adaptive Model Predictive Technique to Control Underactuated Robot and Minimize Energy Consumption. *Procedia CIRP*, 40:407–412, 1 2016. doi:10.1016/J.PROCIR.2016.01.080.

# S1 NSCLC Model

The model of the NSCLC is [27]

$$\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}), \qquad \mathbf{y} = \mathbf{C}\mathbf{x} = [0,0,0,0,0,0,0,1,0,0,0,1,0,0,0,0,0,0,0,0,0]\mathbf{x} \tag{S1}$$

where the vector field $f(.,.)$ is detailed in (S4)-(S23), $\mathbf{x}$ is the state vector containing 21 molecule concentrations (Table S1), $\mathbf{u}$ is the input vector $\mathbf{u} = [I_1, I_2, I_3]^T$ (orange in (S4)-(S23), each input acts on both the active and inactive target molecule and therefore appears twice in (S4)-(S23)) and $\mathbf{y}$ is the vector of outputs $y_1 = pERK$ and $y_2 = pAkt$ (blue in (S4)-(S23)).

| $\mathbf{x_i}$ | **Shorthand** | **Molecule** |
|---|---|---|
| $x_1$ | pEGFR | Active epidermal growth factor receptor |
| $x_2$ | DSOS | Deactive SOS |
| $x_3$ | SOS | Son Of Sevenless, a Guanine nucleotide exchange factor |
| $x_4$ | Raf | Raf kinase |
| $x_5$ | pRas | Active Ras, a small GTPase |
| $x_6$ | pMEK | Active Methyl ethyl ketone |
| $x_7$ | ERK | Extracellular-signal-regulated kinase |
| $x_8$ | pERK | Active ERK |
| $x_9$ | pIGF1R | Active Insulin-like growth factor receptor |
| $x_{10}$ | PI3K | Phosphoinositide 3-kinase |
| $x_{11}$ | pPI3K | Active PI3K |
| $x_{12}$ | pAkt | Active Akt |
| $x_{13}$ | Akt | Set of three serine/threonine-specific protein kinases |
| $x_{14}$ | PP2A | Protein phosphatase 2, Kinase inhibitor |
| $x_{15}$ | Ras | A small GTPase |
| $x_{16}$ | pRaf | Active Raf |
| $x_{17}$ | MEK | Methyl Ethyl Ketone |
| $x_{18}$ | RasGAP | GTP hydrolyser of Ras |
| $x_{19}$ | ppRaf | Raf which has been phosphorylated twice |
| $x_{20}$ | P90 | Ribosomal S6 kinase |
| $x_{21}$ | pP90 | Active P90 |

Table S1

## S1.1 Difference Between NSCLC and Wild Type Cells

Mutations present in NSCLC cells can lead to an overexpression of $EGFR$ and $IGF1R$; this is represented by using different initial conditions for $pEGFR$ and $pIGF1R$ [27] and leads to the response shown in Figure 2.

$$\begin{array}{lll} \text{NSCLC Cell:} & pEGFR_0 = 800000\ \mu M & pIGF1R_0 = 400000\ \mu M \\ \text{Wild Type Cell:} & pEGFR_0 = 8000\ \mu M & pIGF1R_0 = 800\ \mu M \end{array} \tag{S2}$$

$$\begin{array}{cccc} SOS_0 = 120000\ \mu M & Ras_0 = 120000\ \mu M & Raf_0 = 120000\ \mu M & MEK_0 = 600000\ \mu M \\ ERK_0 = 600000\ \mu M & P90Rsk_0 = 120000\ \mu M & PI3K_0 = 120000\ \mu M & ppRaf_0 = 120000\ \mu M \\ & PP2A_0 = 120000\ \mu M & RasGAP_0 = 120000\ \mu M & \\ \multicolumn{4}{c}{DSOS_0 = pRas_0 = pRaf_0 = pMEK_0 = pERK_0 = pP90Rsk_0 = pPI3K_0 = 0\ \mu M} \end{array} \tag{S3}$$

$$\frac{\mathrm{d}pEGFR}{\mathrm{d}t} = -kf_E pEGFR - k_{SOS:E} pEGFR \frac{DSOS}{KM_{SOS:E} + DSOS} - k_{PI3K:E} pEGFR \frac{PI3K}{KM_{PI3K:E} + PI3K} \tag{S4}$$

$$\frac{\mathrm{d}pIGF1R}{\mathrm{d}t} = -kf_I pIGF1R - k_{SOS:I} pIGF1R \frac{DSOS}{KM_{SOS:I} + DSOS} - k_{PI3K:I} pIGF1R \frac{PI3K}{KM_{PI3K:I} + PI3K} \tag{S5}$$

$$\frac{\mathrm{d}SOS}{\mathrm{d}t} = k_{SOS:E} pEGFR \frac{DSOS}{KM_{SOS:E}+DSOS} + k_{SOS:I} pIGF1R \frac{DSOS}{KM_{SOS:I}+DSOS} - k_{DSOS:P90} pP90Rsk \frac{SOS}{KM_{DSOS:P90}+SOS} - k_{Ras:SOS} SOS \frac{Ras}{KM_{Ras:SOS}+Ras} \tag{S6}$$

$$\frac{\mathrm{d}DSOS}{\mathrm{d}t} = k_{DSOS:pP90} pP90Rsk \frac{SOS}{KM_{DSOS:pP90}+SOS} - k_{SOS:E} pEGFR \frac{DSOS}{KM_{SOS:E}+DSOS} - k_{SOS:I} pIGF1R \frac{DSOS}{KM_{SOS:I}+DSOS} \tag{S7}$$

$$\frac{\mathrm{d}pRas}{\mathrm{d}t} = k_{Ras:SOS} SOS \frac{Ras}{KM_{Ras:SOS}+Ras} - k_{Ras:Gab} RasGAP \frac{pRas}{KM_{Ras:Gab}+pRas} - k_{PI3K:Ras} pRas \frac{PI3K}{KM_{PI3K:Ras}+PI3K} - k_{Raf:Ras} pRas \frac{Raf}{KM_{Raf:Ras}+Raf} \tag{S8}$$

$$\frac{\mathrm{d}Ras}{\mathrm{d}t} = k_{Ras:SOS} SOS \frac{Ras}{KM_{Ras:SOS}+Ras} + k_{Ras:Gab} RasGAP \frac{pRas}{KM_{Ras:Gab}+pRas} \tag{S9}$$

$$\frac{\mathrm{d}pRaf}{\mathrm{d}t} = k_{Raf:Ras} pRas \frac{Raf}{KM_{Raf:Ras}+Raf} - k_{Raf:ppRaf} ppRaf \frac{pRaf}{KM_{Raf:ppRaf}+pRaf} - k_{Raf:Akt} pAkt \frac{pRaf}{KM_{Raf:Akt}+pRaf} - k_{Raf:MEK} pRaf \frac{MEK}{KM_{Raf:MEK}+MEK} \tag{S10}$$

$$\frac{\mathrm{d}Raf}{\mathrm{d}t} = -k_{Raf:Ras} pRas \frac{Raf}{KM_{Raf:Ras}+Raf} + k_{Raf:ppRaf} ppRaf \frac{pRaf}{KM_{Raf:ppRaf}+pRaf} + k_{Raf:Akt} pAkt \frac{pRaf}{KM_{Raf:Akt}+pRaf} \tag{S11}$$

$$\frac{\mathrm{d}Raf}{\mathrm{d}t} = -k_{Raf:Ras} pRas \frac{Raf}{KM_{Raf:Ras}+Raf} + k_{Raf:ppRaf} ppRaf \frac{pRaf}{KM_{Raf:ppRaf}+pRaf} + k_{Raf:Akt} pAkt \frac{pRaf}{KM_{Raf:Akt}+pRaf} \tag{S12}$$

$$\frac{\mathrm{d}MEK}{\mathrm{d}t} = -k_{MEK:Raf} pRaf \frac{MEK}{KM_{MEK:Raf}+MEK} + k_{MEK:PP2A} PP2A \frac{pMEK}{KM_{MEK:PP2A}+pMEK} - kon_3 MEK \frac{I_3}{Km_3+I3} \tag{S13}$$

$$\frac{\mathrm{d}pMEK}{\mathrm{d}t} = k_{MEK:Raf} pRaf \frac{MEK}{KM_{MEK:Raf}+MEK} - k_{MEK:PP2A} PP2A \frac{pMEK}{KM_{MEK:PP2A}+pMEK} - k_{ERK:MEK} pMEK \frac{ERK}{KM_{ERK:MEK}+ERK} - kon_3 pMEK \frac{I_3}{Km_3+I3} \tag{S14}$$

$$\frac{\mathrm{d}pERK}{\mathrm{d}t} = k_{ERK:MEK} pMEK \frac{ERK}{KM_{ERK:MEK}+ERK} - k_{ERK:PP2A} PP2A \frac{pERK}{KM_{ERK:PP2A}+pERK} - k_{DSOS:ERK} pERK \frac{SOS}{KM_{DSOS:ERK}+SOS} - k_{E:ERK} pERK \frac{pEGFR}{KM_{EGFR:ERK}+pEGFR} \tag{S15}$$

$$\frac{\mathrm{d}ERK}{\mathrm{d}t} = -k_{ERK:MEK} pMEK \frac{ERK}{KM_{ERK:MEK}+ERK} + k_{ERK:PP2A} PP2A \frac{pERK}{KM_{ERK:PP2A}+pERK} \tag{S16}$$

$$\frac{\mathrm{d}pP90Rsk}{\mathrm{d}t} = k_{pP90:ERK} pERK \frac{P90Rsk}{KM_{pP90:ERK}+P90Rsk} - k_{DSOS:pP90} pP90Rsk \frac{SOS}{KM_{DSOS:pP90}+SOS} \tag{S17}$$

$$\frac{\mathrm{d}P90Rsk}{\mathrm{d}t} = kd_{pP90Rsk} - k_{pP90:ERK} pERK \frac{P90Rsk}{KM_{pP90:ERK}+P90Rsk} \tag{S18}$$

$$\frac{\mathrm{d}pPI3K}{\mathrm{d}t} = k_{PI3K:Ras}pRas\frac{PI3K}{KM_{PI3K:Ras}+PI3K} + k_{PI3K:I}pIGF1R\frac{PI3K}{KM_{PI3K:I}+PI3K} + k_{PI3K:E}pEGFR\frac{PI3K}{KM_{PI3K:E}+PI3K} - k_{Akt:PI3K}Akt\frac{pPI3K}{KM_{Akt:PI3K}+pPI3K} - kd_{PI3K}pPI3K - kon_1pPI3K\frac{I_1}{Km_1+I1} \tag{S19}$$

$$\frac{\mathrm{d}PI3K}{\mathrm{d}t} = -k_{PI3K:Ras}pRas\frac{PI3K}{KM_{PI3K:Ras}+PI3K} - k_{PI3K:I}pIGF1R\frac{PI3K}{KM_{PI3K:I}+PI3K} - k_{PI3K:E}pEGFR\frac{PI3K}{KM_{PI3K:E}+PI3K} + kd_{PI3K}pPI3K - kon_1PI3K\frac{I_1}{Km_1+I1} \tag{S20}$$

$$\frac{\mathrm{d}pAkt}{\mathrm{d}t} = -k_{Raf:Akt}pAkt\frac{pRaf}{KM_{Raf:Akt}+pRaf} + k_{Akt:PI3K}Akt\frac{pPI3K}{KM_{Akt:PI3K}+pPI3K} - kon_2pAkt\frac{I_2}{Km_2+I2} \tag{S21}$$

$$\frac{\mathrm{d}Akt}{\mathrm{d}t} = -k_{Akt:PI3K}Akt\frac{pPI3K}{KM_{Akt:PI3K}+pPI3K} + kd_{Akt}pAkt - kon_2Akt\frac{I_2}{Km_2+I2} \tag{S22}$$

$$\frac{\mathrm{d}RasGAP}{\mathrm{d}t} = 0 \quad \frac{\mathrm{d}ppRaf}{\mathrm{d}t} = 0 \quad \frac{\mathrm{d}PP2A}{\mathrm{d}t} = 0 \tag{S23}$$

## S1.2 Conservation Equations

Each molecule can either be active, $p(.)$, or inactive, but there is a constant total concentration of each molecule within the model. This total, $(.)_T$, is defined by the following conservation equations:

$$\begin{array}{ccc} EGFR_T = pEGFR + EGFR & IGF1R_T = pIGF1R + IGF1R & SOS_T = SOS + DSOS \\ Ras_T = pRas + Ras & Raf_T = pRaf + Raf & MEK_T = pMEK + MEK \\ ERK_T = pERK + ERK & PI3K_T = pPI3K + PI3K & Akt_T = pAkt + Akt \end{array} \tag{S24}$$

## S1.3 Parameters in the NSCLC Model

| Variable | Definition | Value, [27] |
|---|---|---|
| $kf_E$ | EGFR deactivation | $0.02h^{-1}$ |
| $k_{SOS:E}$ | Catalytic constant for SOS activation by EGFR | 694.731 $min^{-1}$ |
| $KM_{SOS:E}$ | Michaelis–Menten constant for SOS activation by EGFR | 6086070.0 $\mu M$ |
| $k_{PI3K:E}$ | Catalytic constant for PIK3 activation by EGFR | 10.6737 $min^{-1}$ |
| $KM_{PI3K:E}$ | Michaelis–Menten constant for PIK3 activation by EGFR | 184912.0 $\mu M$ |
| $kf_I$ | IGF1R deactivation | 0.02 $min^{-1}$ |
| $k_{SOS:I}$ | Catalytic constant for SOS activation by IGF1R | 500.0 $min^{-1}$ |
| $KM_{SOS:I}$ | Michaelis–Menten constant for SOS activation by IGF1R | 1000000.0 $\mu M$ |
| $k_{PI3K:I}$ | Catalytic constant for PIK3 activation by IGF1R | 10.6737 $min^{-1}$ |
| $KM_{PI3K:I}$ | Michaelis–Menten constant for PIK3 activation by IGF1R | 184912.0 $\mu M$ |
| $k_{DSOS:P90}$ | Catalytic constant for DSOS deactivation by p90Rsk | 161197.0 $min^{-1}$ |
| $KM_{DSOS:P90}$ | Michaelis–Menten constant for DSOS deactivation by p90Rsk | 896896.0 $\mu M$ |
| $k_{Ras:SOS}$ | Catalytic constant for Ras activation by SOS | 32.344 $min^{-1}$ |
| $KM_{Ras:SOS}$ | Michaelis–Menten constant for Ras activation by SOS | 35954.3 $\mu M$ |
| $k_{Ras:Gab}$ | Catalytic constant for Ras deactivation by RasGAP | 1509.36 $min^{-1}$ |
| $KM_{Ras:Gab}$ | Michaelis–Menten constant for Ras deactivation by RasGAP | 1432410.0 $\mu M$ |
| $k_{Raf:Ras}$ | Catalytic constant for Raf activation by Ras | 0.884096 $min^{-1}$ |
| $KM_{Raf:Ras}$ | Michaelis–Menten constant for Raf deactivation by Ras | 62464.6 $\mu M$ |
| $k_{Raf:ppRaf}$ | Catalytic constant for Raf deactivation by RafPP | 0.126329 $min^{-1}$ |
| $KM_{Raf:ppRaf}$ | Michaelis–Menten constant for Raf deactivation by RafPP | 1061.71 $\mu M$ |
| $k_{Raf:Akt}$ | Catalytic constant for Raf deactivation by Akt | 15.1212 $min^{-1}$ |
| $KM_{Raf:Akt}$ | Michaelis–Menten constant for Raf deactivation by Akt | 119355.0 $\mu M$ |
| $k_{Raf:MEK}$ | Catalytic constant for MEK activation by Raf | 185.759 $min^{-1}$ |
| $KM_{Raf:MEK}$ | Michaelis–Menten constant for MEK activation by Raf | 4768350.0 $\mu M$ |
| $k_{MEK:PP2A}$ | Catalytic constant for MEK deactivation by PP2A | 2.83243 $min^{-1}$ |
| $KM_{MEK:PP2A}$ | Michaelis–Menten constant for MEK deactivation by PP2A | 518753.0 $\mu M$ |
| $k_{ERK:MEK}$ | Catalytic constant for ERK activation by MEK | 9.85367 $min^{-1}$ |
| $KM_{ERK:MEK}$ | Michaelis–Menten constant for ERK deactivation by MEK | 1007340.0 $\mu M$ |
| $k_{ERK:PP2A}$ | Catalytic constant for ERK activation by PP2A | 9.85367 $min^{-1}$ |
| $KM_{ERK:PP2A}$ | Michaelis–Menten constant for ERK deactivation by PP2A | 1007340.0 $\mu M$ |
| $kd_{P90}$ | p90Rsk deactivation | 0.0050 $min^{-1}$ |
| $k_{P90:ERK}$ | Catalytic constant for p90Rsk activation by Erk | 0.0213697 $min^{-1}$ |
| $KM_{P90:ERK}$ | Michaelis–Menten constant for p90Rsk activation by Erk | 763523.0 $\mu M$ |
| $k_{Akt:PI3K}$ | Catalytic constant for Akt activation by PIK3 | 0.0566279 $min^{-1}$ |
| $KM_{Akt:PI3K}$ | Michaelis–Menten constant for Akt activation by PIK3 | 653951.01 $\mu M$ |
| $kd_{Akt}$ | Akt deactivation | 0.0050 $min^{-1}$ |
| $kd_{PI3K}$ | PI3K deactivation | 0.0050 $min^{-1}$ |
| | **Value estimated in this report** | |
| $Kon_1$ | Catalytic constant for PI3K deactivation by $I_1$ | 0.1 $min^{-1}$ |
| $Km_1$ | Michaelis-Menten constant for PI3K deactivation by $I_1$ | 60 $\mu M$ |
| $Kon_2$ | Catalytic constant for Akt deactivation by $I_2$ | 0.01 $min^{-1}$ |
| $Km_2$ | Michaelis-Menten constant for Akt deactivation by $I_2$ | 8.9 $\mu M$ |
| $Kon_3$ | Catalytic constant for ERK deactivation by $I_3$ | 2 $min^{-1}$ |
| $Km_3$ | Michaelis-Menten constant for ERK deactivation by $I_3$ | 2.5 $\mu M$ |

Table S2: Parameters used in the NSCLC model including the six input parameters discussed in Section S2.

## S2 Parameter Choice For Input Interactions

The equation governing the inputs reaction with the target molecule is given by

$$\frac{\mathrm{d}Target}{\mathrm{d}t} = -Target\frac{KonI}{Km + I}, \tag{S25}$$

where the Michealis-Menten parameter values associated with each input are reported in Table S3. $Km$ is equivalent to the $IC_{50}$ value of the inhibitor being used on each target. To make the response of the controller inputs less 'switch like', inhibitors with relatively large $IC_{50}$ values have been chosen. $Kon$ has then been chosen by fitting the model simulations to western blots for $I_1$-3MA [52], $I_2$-Oridonin [53] and $I_3$-Pimasetib [54].

| Drug | $Km(\mu M)$ | $Kon(min^{-1})$ | Target Molecule |
|---|---|---|---|
| $I_1$ - 3MA | 60 [55] | 0.1 | PI3K |
| $I_2$ - Oridonin | 8.9 [56] | 0.01 | Akt |
| $I_3$ - Pimasetib | 2 [57] | 2.5 | MEK |

Table S3: Input properties used in the simulations

## S3 Step Input Response

Step input simulations for each input, as described in Section 3.1.1, can be seen in Figure S1. In all cases, the control performance is worst as compared to that obtained with SISO control (Figure 4), as the $EI$ is either comparable (S1A) or higher (S1C and E) than those obtained applying feedback control; also, in all step input simulations, the Dose Index is higher.

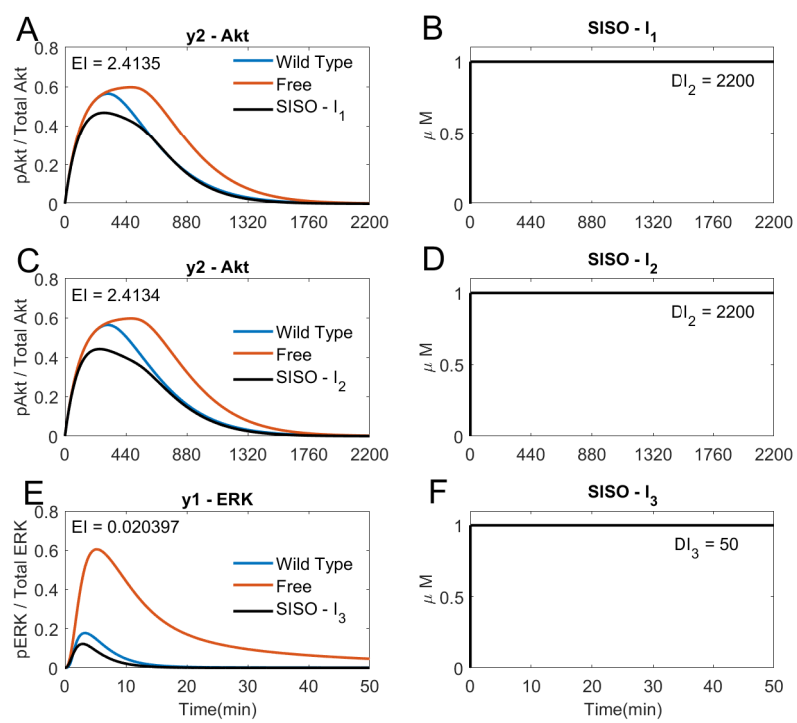


Figure S1: Step Input simulations. (A) The response of y2 (Akt) to the step input of $I_1$ in (B). (C) The response of y2 (Akt) to the step input of $I_2$ in (D). (E) The response of y1 (ERK) to the step input of $I_3$ in (F).

# S4 Model Predictive Control (MPC)

MPC uses a model of the plant (system to be controlled) in the feedback loop to estimate the effect of the inputs the controller will choose at each time step. The inputs are chosen to minimise a user-defined cost function that typically includes terms penalising the magnitude of the inputs, $\mathbf{u}(t)$, and the magnitude of errors, $\mathbf{e}(t)$, between the response of the system and reference signals [29], as shown in Figure S2. The feedback loop then measures the outputs of the actual plant, $\mathbf{y}(t)$, to estimate the actual states, $\mathbf{x}(t)$, and reiterates the MPC scheme to choose each subsequent input, $\mathbf{u}(t)$.

Some modifications to the cost function are discussed in Sections S4.2 and S4.3. MPC controllers are not limited to linear systems, however, non-linear systems can result in a larger computational effort and require more complex optimisation solvers, as discussed in Section S7.

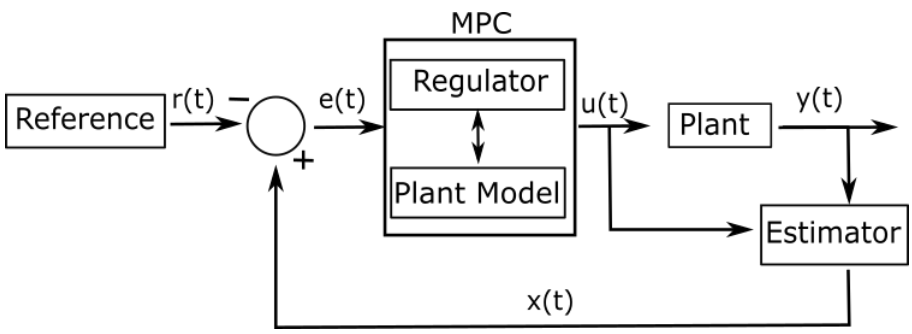


Figure S2: MPC architecture.

For the present numerical study, the plant is the nonlinear NSCLC model presented in Equations (S4)–(S23) and all the system states, $\mathbf{x}(t)$, are measured directly. In practice, only a few outputs/states can be measured and an estimator is required to estimate the remaining states from input-output data as shown in Figure S2.

The control reference signal, $\mathbf{r}(t)$, is chosen as the response of a wild type cell (i.e. without cancer). $\mathbf{e}(t)$ is the error signal between the reference, $\mathbf{r}(t)$, and internal states of the plant, $\mathbf{x}(t)$. $\mathbf{e}(t)$ is fed into the regulation block of the control scheme; this is where the optimisation problem is solved.

## S4.1 Cost Function Derivation

The internal state errors $\mathbf{e}(t)$ are calculated and fed into the MPC block at each time step. The MPC controller uses the model of the plant system to predict the future state error of the system for possible combinations of inputs, within the problem constraints, over the prediction horizon, $N$. The controller then optimally chooses the input profile that results in the minimum of a predetermined cost function, $J(\mathbf{U})$, in the regulator. The inputs of the first time step of this optimal sequence are then applied to the plant system. At the next time step, the error in the states is estimated and this process repeats.

Usually, the optimisation problem contains the cost function to be minimised, and the state and input constraints.

$$\begin{aligned} \min_{\mathbf{U}}(J(\mathbf{E},\mathbf{U})) \quad st. \quad -\mathbf{U} \leq \mathbf{U}_L, \quad \mathbf{U} \leq \mathbf{U}_U, \\ \text{where} \quad J(\mathbf{E},\mathbf{U}) = \mathbf{E}^T\mathbf{Q}\mathbf{E} + \mathbf{U}^T\mathbf{R}\mathbf{U} \end{aligned} \tag{S26}$$

The model that the regulator sees is a discrete approximations of the NSCLC model for $1 \leq k \leq N$ steps. $\mathbf{E} = [\mathbf{e}(0), \mathbf{e}(1), \ldots, \mathbf{e}(N)]^T$ describes the current and future predicted state errors. $\mathbf{U} = [\mathbf{u}(1), \mathbf{u}(2), \ldots, \mathbf{u}(N)]^T$ describes the future inputs. The weight of the cost function related to each term can vary what is considered the optimal input. $\mathbf{Q}$ weights the error of the states and $\mathbf{R}$ weights the use of the inputs [29]. For example, if we

want to minimise the use of a drug then the weight of the cost function associated with the inputs, $\mathbf{R}$, should be relatively large compared to the weight of the state error, $\mathbf{Q}$. Constraints on the input bounds are also included here ($\mathbf{U}_L \leq \mathbf{U} \leq \mathbf{U}_U$, in all simulations $\mathbf{U}_L = 0\mu M$ and $\mathbf{U}_U = 1\mu M$).

### S4.1.1 Linear MPC

If a linear approximation of the model can be produced and represented by a state space, the future behaviour of the model can be calculated offline and the optimisation problem is convex (for affine constraints). The cost function can be minimised using a quadratic solver, which is relatively computationally light (MATLAB R2021b's 'quadprog' solver was used here). Here, a state space was used to represent the plant's model:

$$\mathbf{e}(k+1) = \mathbf{A}\mathbf{e}(k) + \mathbf{B}\mathbf{u}(k+1) \qquad \mathbf{y}(k) = \mathbf{C}\mathbf{e}(k) \tag{S27}$$

As long as the current state is known, each future state can be estimated for a given input profile, i.e.

$$\begin{aligned} \mathbf{e}(1) &= \mathbf{A}\mathbf{e}_0 + \mathbf{B}\mathbf{u}(1) \\ \mathbf{e}(2) &= \mathbf{A}\mathbf{e}_1 + \mathbf{B}\mathbf{u}(2) = \mathbf{A}(\mathbf{A}\mathbf{e}_0 + \mathbf{B}\mathbf{u}(1)) + \mathbf{B}\mathbf{u}(2) \\ \mathbf{e}(N) &= \mathbf{A}^N\mathbf{e}_0 + \mathbf{A}^{N-1}\mathbf{B}\mathbf{u}(1) + \mathbf{A}^{N-2}\mathbf{B}\mathbf{u}(2) + \ldots + \mathbf{B}\mathbf{u}(N). \end{aligned} \tag{S28}$$

Defining a new notation containing all of the states in the prediction horizon we get:

$$\mathbf{E} = \mathbf{M}\mathbf{e}_0 + \tilde{\mathbf{C}}\mathbf{U}$$

$$\mathbf{U} = \begin{bmatrix} \mathbf{u}(1) \\ \mathbf{u}(2) \\ \vdots \\ \mathbf{u}(N) \end{bmatrix} \qquad \mathbf{M} = \begin{bmatrix} \mathbf{I} \\ \mathbf{A} \\ \vdots \\ \mathbf{A}^N \end{bmatrix} \qquad \tilde{\mathbf{C}} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \ldots & \mathbf{0} \\ \mathbf{B} & \mathbf{0} & \ldots & \mathbf{0} \\ \mathbf{AB} & \mathbf{B} & \ldots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{A}^{N-1}\mathbf{B} & \mathbf{A}^{N-2}\mathbf{B} & \ldots & \mathbf{B} \end{bmatrix} \tag{S29}$$

Therefore, the cost function can be rearranged as

$$\begin{aligned} J(\mathbf{E}(\mathbf{U}), \mathbf{U}) &= \mathbf{E}^T\mathbf{Q}\mathbf{E} + \mathbf{U}^T\mathbf{R}\mathbf{U} \\ J(\mathbf{U}) &= (\mathbf{M}\mathbf{e}_0 + \tilde{\mathbf{C}}\mathbf{U})^T\mathbf{Q}(\mathbf{M}\mathbf{e}_0 + \tilde{\mathbf{C}}\mathbf{U}) + \mathbf{U}^T\mathbf{R}\mathbf{U} \\ &= \mathbf{U}^T(\tilde{\mathbf{C}}^T\mathbf{Q}\tilde{\mathbf{C}} + \mathbf{R})\mathbf{U} + 2\mathbf{e}_0^T(\tilde{\mathbf{C}}^T\mathbf{Q}\mathbf{M})^T\mathbf{U} + \mathbf{e}_0^T\mathbf{M}^T\mathbf{Q}\mathbf{M}\mathbf{e}_0 \\ &= \mathbf{U}^T(\tilde{\mathbf{Q}} + \mathbf{R})\mathbf{U} + \tilde{\mathbf{q}}\mathbf{U} + \mathbf{e}_0^T\mathbf{M}^T\mathbf{Q}\mathbf{M}\mathbf{e}_0 \end{aligned} \tag{S30}$$

The final term in the cost function can be removed as it is constant with respect to the inputs and therefore will not affect the position of the minimum point; thus,

$$\begin{aligned} J(\mathbf{U}) &= \mathbf{U}^T\mathbf{R}\mathbf{U} + \left(\mathbf{U}^T\tilde{\mathbf{Q}}\mathbf{U} + \tilde{\mathbf{q}}\mathbf{U}\right) \\ &= \mathbf{U}^T(\tilde{\mathbf{Q}} + \mathbf{R})\mathbf{U} + \tilde{\mathbf{q}}\mathbf{U} \end{aligned} \tag{S31}$$

Only the optimal inputs for the first time step are then applied to the plant, $\mathbf{u}(1)$, the whole optimisation process is repeated at the next time step.

### S4.1.2 Weighting of the Traditional Cost Function

The cost function as shown in (S30), can be weighted as a balance of using the inputs, $\gamma$; the error in all of the estimated states, $\alpha$ and the error in the outputs, $\beta$.

$$\mathbf{Q} = \alpha\mathbf{I} + \beta \begin{bmatrix} \mathbf{C}^T\mathbf{C} & \dots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \dots & \mathbf{C}^T\mathbf{C} \end{bmatrix}$$
$$\mathbf{R} = \begin{bmatrix} \gamma\mathbf{I} & \dots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \dots & \gamma\mathbf{I} \end{bmatrix} \tag{S32}$$

where $\mathbf{I}$ is an identity matrix. The cost function can be tuned by $\alpha$, $\beta$ and $\gamma$.

$$J(\mathbf{U}) = \mathbf{U}^T(\tilde{\mathbf{Q}}(\alpha,\beta) + \mathbf{R}(\gamma))\mathbf{U} + \tilde{\mathbf{q}}(\alpha,\beta)\mathbf{U} \tag{S33}$$

## S4.2 Differential Terms in the Cost Function

The controller can favour to rapidly change input concentrations, which is not ideal for *in vitro* experiments, where there might be delays in the actuation, and frequent media change might cause stress to cells. Therefore, a term related to the gradient of the inputs was added to the cost function to reduce fast variations of the inputs. The linear approximation of the model used within the MPC simulations is discrete and therefore the derivative is approximated by the scaled difference between inputs at adjacent time steps.

$$\frac{\mathrm{d}\mathbf{U}}{\mathrm{d}t}|_k \approx \frac{\mathbf{U}(k) - \mathbf{U}(k-1)}{Ts} \tag{S34}$$

Using the squared sum of the derivative of the inputs, the gradient of the steps between the last actual input and the last step in the prediction horizon can be added to the cost function. Any constant scaling can be dropped as this would just change the weighting added to the term.

$$\begin{aligned} \textstyle\sum_{k=j}^{k=j+N}(\mathbf{U}(k) - \mathbf{U}(k-1))^2 &= \textstyle\sum_{k=j}^{k=j+N} \mathbf{U}(k)^2 + \mathbf{U}(k-1)^2 - 2\mathbf{U}(k)\mathbf{U}(k-1) \\ &= \mathbf{U}(j-1)^2 - 2\mathbf{U}(j)\mathbf{U}(j-1) + 2[\mathbf{U}(j)^2 + ... + \mathbf{U}(j+N-1)^2 \\ &\quad -\mathbf{U}(j)\mathbf{U}(j+1) - ... - \mathbf{U}(j+N-1)\mathbf{U}(j+N)] + \mathbf{U}(j+N)^2 \\ &= \mathbf{U}^T\mathbf{D}\mathbf{U} + \mathbf{d}\mathbf{U} \end{aligned} \tag{S35}$$

The gradient can be added to the cost function, $J(\mathbf{U})$, (Equation S33,) through $\mathbf{D}$ and $\mathbf{d}$, pre-multiplying by $\theta$.

$$\mathbf{D} = \theta \begin{bmatrix} 2\mathbf{I} & -\mathbf{I} & \mathbf{0} & \dots & \mathbf{0} \\ -\mathbf{I} & 2\mathbf{I} & -\mathbf{I} & \dots & \mathbf{0} \\ \mathbf{0} & -\mathbf{I} & 2\mathbf{I} & \dots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{I} \end{bmatrix} \quad \mathbf{d} = \theta \begin{bmatrix} 2\mathbf{U}(j-1) & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} \end{bmatrix} \tag{S36}$$

where $\mathbf{I}$ is a square identity matrix (size equal to the number of inputs).

$$\begin{aligned} J(\mathbf{U}) &= \mathbf{U}^T\mathbf{R}\mathbf{U} + \left(\mathbf{U}^T\tilde{\mathbf{Q}}\mathbf{U} + \tilde{\mathbf{q}}\mathbf{U}\right) + \left(\mathbf{U}^T\mathbf{D}\mathbf{U} + \mathbf{d}\mathbf{U}\right) \\ &= \mathbf{U}^T(\tilde{\mathbf{Q}}(\alpha,\beta) + \mathbf{R}(\gamma) + \mathbf{D}(\theta))\mathbf{U} + (\tilde{\mathbf{q}}(\alpha,\beta) + \mathbf{d}(\theta))\mathbf{U} \end{aligned} \tag{S37}$$

Figure 3 demonstrates the effect of the differential cost.

An alternative approach would have been to add the gradient as a constraint into the optimisation problem preventing the inputs from varying faster than a limiting value. However, this would have merely limited the maximum rate of inputs' variation.

## S4.3 Integral Terms in the Cost Function

When controlling a signalling phosphorylation cascades, it is desirable to decrease both the peak and the duration of an error [30]. Both the duration and the peak are included in the integral of the outputs. Therefore the integral of the output errors should be added as a term in the cost function. The square of the integral errors has been approximated.

$$\begin{aligned}
\left(\int_j^{j+N} \mathbf{e}(t)dt\right)^2 &\approx \sum_{k=j}^{k=j+N} \mathbf{e}(k)^2 + \mathbf{e}(k-1)^2 + 2\mathbf{e}(k)\mathbf{e}(k-1) \\
&\approx \mathbf{e}(j-1)^2 + 2\mathbf{e}(j-1)\mathbf{e}(j) + \mathbf{e}(j+N)^2 + \\
&\quad 2\Big(\mathbf{e}(j)^2 + \ldots + \mathbf{e}(j+N-1)^2 + \mathbf{e}(j)\mathbf{e}(j+1) + \ldots + \mathbf{e}(j+N-1)\mathbf{e}(j+N)\Big) \\
&\approx \mathbf{E}^T\mathbf{P}\mathbf{E} + \mathbf{p}\mathbf{E}
\end{aligned} \tag{S38}$$

The first term is constant and is therefore dropped. $\mathbf{P}$ and $\mathbf{p}$ are weighted by $\eta$ in the cost function.

$$\mathbf{P} = \eta \begin{bmatrix} 2\mathbf{C}^T\mathbf{C} & \mathbf{C}^T\mathbf{C} & \mathbf{0} & \ldots & \mathbf{0} \\ \mathbf{C}^T\mathbf{C} & 2\mathbf{C}^T\mathbf{C} & \mathbf{C}^T\mathbf{C} & \ldots & \mathbf{0} \\ \mathbf{0} & \mathbf{C}^T\mathbf{C} & 2\mathbf{C}^T\mathbf{C} & \ldots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \ldots & \mathbf{C}^T\mathbf{C} \end{bmatrix} \quad \mathbf{p} = \eta \begin{bmatrix} \mathbf{0} & 2\mathbf{C}^T\mathbf{C}\mathbf{e}_0^T & \mathbf{0} & \ldots & \mathbf{0} \end{bmatrix} \tag{S39}$$

The integral cost is defined in terms of the future state errors $\mathbf{E}$ rather than the inputs $\mathbf{U}$. The matrices defined in Equation (S29) can be used.

$$\begin{aligned}
\mathbf{E}^T\mathbf{P}\mathbf{E} + \mathbf{p}\mathbf{E} &= (\mathbf{M}\mathbf{e}_0 + \tilde{\mathbf{C}}\mathbf{U})^T\mathbf{P}(\mathbf{M}\mathbf{e}_0 + \tilde{\mathbf{C}}\mathbf{U}) + \mathbf{p}(\mathbf{M}\mathbf{e}_0 + \tilde{\mathbf{C}}\mathbf{U}) \\
&= \mathbf{e}_0^T\mathbf{M}^T\mathbf{P}\mathbf{M}\mathbf{e}_0 + \mathbf{U}^T\tilde{\mathbf{C}}^T\mathbf{P}\tilde{\mathbf{C}}\mathbf{U} + 2\mathbf{e}_0^T\mathbf{M}^T\mathbf{P}\tilde{\mathbf{C}}\mathbf{U} + \mathbf{p}\mathbf{M}\mathbf{e}_0 + \mathbf{p}\tilde{\mathbf{C}}\mathbf{U}
\end{aligned} \tag{S40}$$

The constant terms with respect to $\mathbf{U}$ can be dropped and the quadratic and proportional terms reorganised.

$$\begin{gathered}
\mathbf{U}^T\tilde{\mathbf{C}}^T\mathbf{P}\tilde{\mathbf{C}}\mathbf{U} + 2\mathbf{e}_0^T\mathbf{M}^T\mathbf{P}\tilde{\mathbf{C}}\mathbf{U} + \mathbf{p}\tilde{\mathbf{C}}\mathbf{U} = \mathbf{U}^T\tilde{\mathbf{P}}\mathbf{U} + \tilde{\mathbf{p}}\mathbf{U} \\
\tilde{\mathbf{P}} = \tilde{\mathbf{C}}^T\mathbf{P}\tilde{\mathbf{C}} \quad \tilde{\mathbf{p}} = \mathbf{p}\tilde{\mathbf{C}} + (2\mathbf{e}_0^T\mathbf{M}^T\mathbf{P}\tilde{\mathbf{C}})
\end{gathered} \tag{S41}$$

The cost function, $J(\mathbf{U})$ (from Equation S37), can be formed including the integral of the state error.

$$\begin{aligned}
J(\mathbf{U}) &= \mathbf{U}^T\mathbf{R}\mathbf{U} + \left(\mathbf{U}^T\tilde{\mathbf{Q}}\mathbf{U} + \tilde{\mathbf{q}}\mathbf{U}\right) + \left(\mathbf{U}^T\mathbf{D}\mathbf{U} + \mathbf{d}\mathbf{U}\right) + \left(\mathbf{U}^T\tilde{\mathbf{P}}\mathbf{U} + \tilde{\mathbf{p}}\mathbf{U}\right) \\
&= \mathbf{U}^T(\tilde{\mathbf{Q}}(\alpha,\beta) + \mathbf{R}(\gamma) + \mathbf{D}(\theta) + \tilde{\mathbf{P}}(\eta))\mathbf{U} + (\tilde{\mathbf{q}}(\alpha,\beta) + \mathbf{d}(\theta) + \tilde{\mathbf{p}}(\eta))\mathbf{U}
\end{aligned} \tag{S42}$$

This is the cost function that has been used in all the simulations with different weights. Figure 3A) demonstrates the effect of the integral cost on reducing both the amplitude and the duration of the outputs.

## S4.4 Adaptive MPC

Adaptive MPC describes an MPC control scheme in which the model of the plant changes as the simulation progresses [58]. Here, it is assumed that the states of the actual plant can be perfectly measured at each step (as no estimator is used). Our non-linear model of the NSCLC, (S4) - (S23), can then be linearised about these current measurements of the states, $\mathbf{x}(k)$, at each time step. This linear model forms the state space in Equation S27 and the cost function is reformed at each iteration to better represent the local future dynamics of the plant. The adaptive linear MPC has a better performance than a single linear model and is less computationally expensive than using a full non-linear model [28, 59, 60].

## S5 Normalisation of Indexes And Bliss Independence

In order to easily compare simulations, it is useful to have an index which summarises the performance of the controller and the type of input it takes to achieve this performance, $EI$ and $DI_i$, respectively (normalised here such that multiple plots can be compared within Figure 7).

The Error Index, $EI$, is the sum of the squared errors of the outputs, calculated by integrating the square of all the output error signals by using a trapezium approximation of the discrete data:

$$EI = \int_0^T \mathbf{e}^T \mathbf{C} \mathbf{e} \, dt \tag{S43}$$

The largest $EI$ (worst performance) found in any MPC simulation using the chosen cost function to control y2 (Akt) is the SISO simulation using $I_1$ ($EI = 2.75$, Figure 4). This has been used to normalise $EI$ to form $\hat{EI}$.

$$\hat{EI} = \frac{EI}{\max\limits_{\gamma_2/\gamma_1}(EI)} = \frac{EI}{EI_{I_1 SISO}} \tag{S44}$$

$DI_i$ is equivalent to the integration of input profile for each input, $I_i$:

$$DI_i = \int_0^T I_i(t) \, dt, \quad DI_{iSISO} = \int_0^T I_i^{SISO}(t) \, dt \tag{S45}$$

$DI_i$ is normalised by the $DI_i$ of $I_i$'s SISO simulation ($I_1$ and $I_2$ acting on y2 (Akt) and $I_3$ acting on y1 (ERK) in Figure 4, $DI_{1SISO} = 1068$, $DI_{2SISO} = 546$, $DI_{3SISO} = 4$), producing $\hat{DI}_i$.

$$\hat{DI}_i = \frac{DI_i}{DI_{iSISO}} \tag{S46}$$

$\hat{DI}_i$ does not give a quantitative measure on the dose of the combined input profile. Within current literature, there are many methods of trying to summarises the joint effect and toxicity of combination therapies, where multiple drugs are given together at a determined time point [32]; however these do not look into dynamic dosages over a given time period. Therefore a combined effect of the drug profiles can be estimated by replacing these static drug dosages with the normalised Dose Index, $\hat{DI}_i$.

An Isobole can be defined as $Iso = \hat{DI}_1 + \hat{DI}_2$ for these therapies. From this definition our combinations are all antagonistic. The Bliss Independence formula, $BI$, assumes that there is no correlation between the two agents.

$$BI = \hat{DI}_1 + \hat{DI}_2 - \hat{DI}_1 \hat{DI}_2 \tag{S47}$$

Our model is deterministic, with each input having a different target molecule, therefore within these *in silico* simulations there is no correlation between the inputs. Therefore the Bliss Independence formula can be used to gauge the combined effect of 2 drugs [33]. All three indexes are used in Figure 7 to compare multiple simulations.

## S6 Discrete Drug Holiday

A discrete MISO simulation, as described in Section 3.4, where $T_s = 1min$ can be seen in Figure S3. Figure S3 achieves an $EI = 1.3533$, significantly better than just a SISO simulation in Figure 4 ($EI \rightarrow 1.35 < 1.95 < 2.75$) whilst using a lower Dose Index ($DI_1 \rightarrow 645 < 1068$ and $DI_2 \rightarrow 438 < 546$). However in this simulation both inputs rapidly fluctuate from $0\mu M$ to $1\mu M$, therefore a longer time step can be used to reduce the fluctuations.

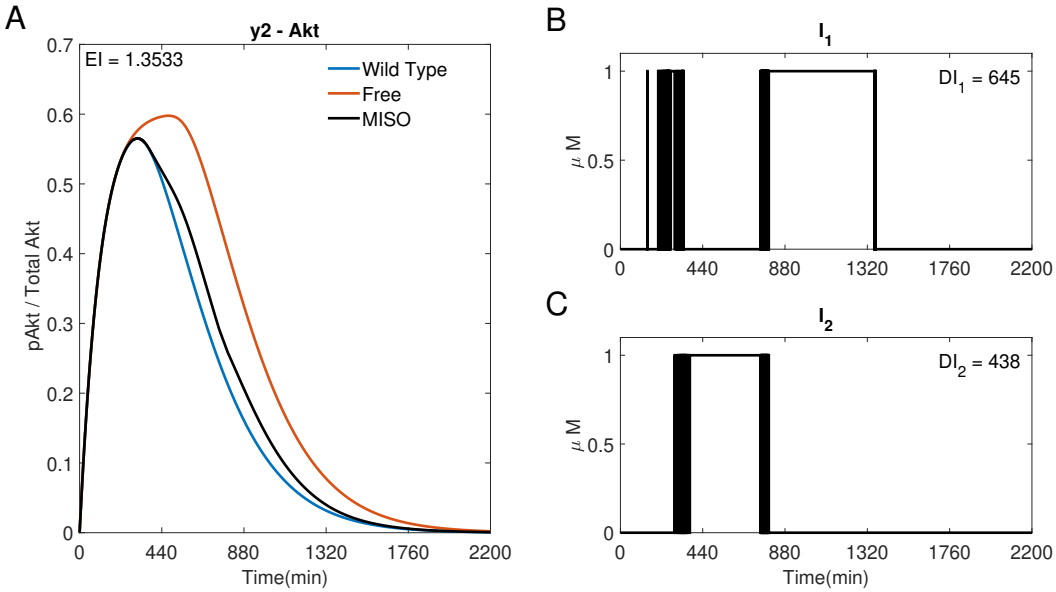


Figure S3: A discrete MISO adaptive MPC simulation using $I_1$ and $I_2$ to control the concentration of y2(Akt). A) The response of Akt. B) and C) The inputs used in the simulation. Parameters: $T_s = 1min$, $N = 10$, $\alpha = 0$, $\beta = 0$, $\gamma = [1, 10^5, -]$, $\theta = 0$ and $\eta = 1$.

# S7 Linear vs Non-Linear MPC

All MPC simulations use an adaptive linear MPC controller, where the linear model is based off a linearisation of a non-linear model of the NSCLC system (S4)-(S23). Using non-linear MPC creates a non-convex optimisation problem requiring a more complex (and computationally heavy) solver. The non-linear simulations have used MATLAB's 'fmincon', a gradient based non-linear solver. It is the fastest appropriate solver in MATLAB R2021b.

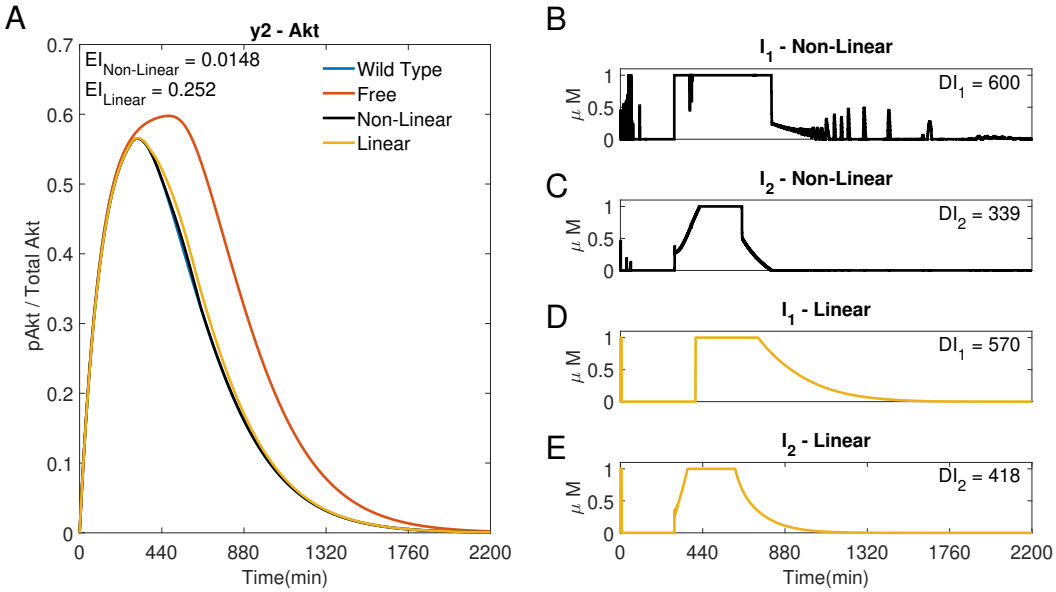


Figure S4: Two MISO MPC simulations using $I_1$ and $I_2$ to control the concentration of y2 (Akt). (A) The response of Akt to a non-linear MPC controller∼ and an adaptive linear MPC controller ∼. (B, C) Inputs $I_1$ and $I_2$ used in the non-linear simulation. (D, E) Inputs $I_1$ and $I_2$ used in the linear simulation. Parameters: $T_s = 1$, $N = 10$, $\alpha = 0$, $\beta = 0$, $\gamma = [1, 10^5, -]$, $\theta = 0$ and $\eta = 1$.

Figure S4 compares a MISO response using adaptive linear MPC ∼ to non-linear MPC ∼. It can be seen that the non-linear MPC has a lower Error Index of $EI = 0.0148$ compared to the adaptive linear MPC's $EI = 0.2520$. However, the non-linear MPC had a significantly higher run-time, as expected.

Non-linear MPC would limit the controller's use *in vitro*, as it the time to process the measurements might be longer than the data acquisition sampling time. This would then suggest using a larger sampling time, possibly causing issues with the controller performance (see Figures 5 and 6). When using the adaptive linear MPC controller, each iteration of the algorithm is well within the sampling time and enables capturing key dynamics of the system even though some of the non-linear couplings between the states are lost.

## S8 MPC vs Proportional (P) Control Schemes

All feedback simulations have used an MPC controller. Figure S5 compares the performance of a linear adaptive MPC controller ∼ to a Proportional controller ∼. Due to the relatively slow changing outputs, a differential gain was not used. An integral gain is not used as the output concentration is almost always greater than the reference, therefore the integral error never resets to zero, leaving the inputs at a non-zero steady state, causing a high $DI_i$. Therefore a Proportional (P) controller is used. The two gains (for each input) can be tuned such that the P controller's initial reaction to the state error results in a relatively low $EI$, as shown in Figure S5 ∼, however the response is sensitive to the choice of gains. It can be seen in Figure S5 B) and C) that the inputs are identical, as the controller does not know the dynamics of the plant.

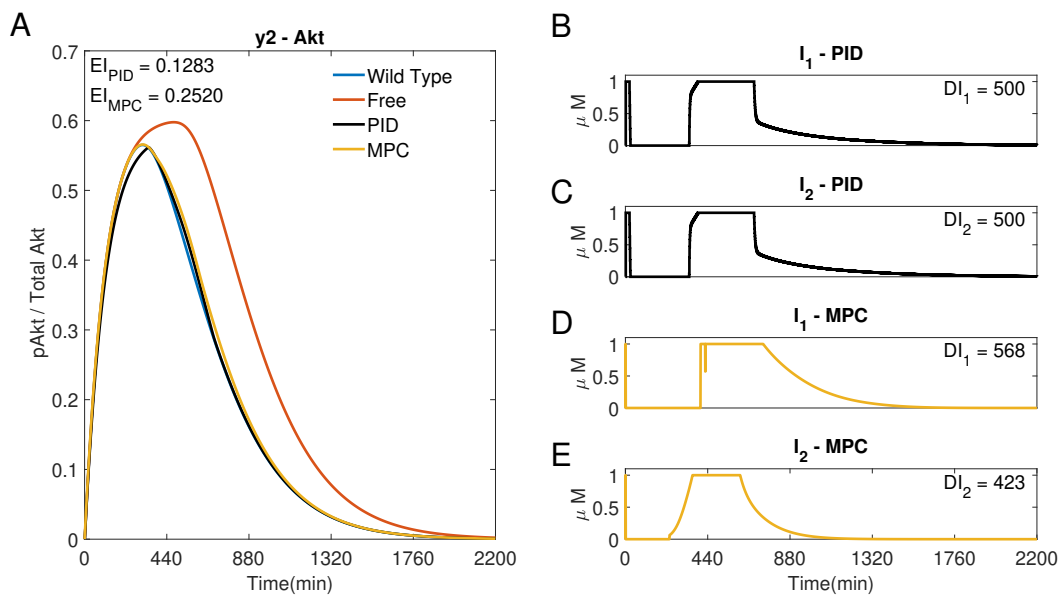


Figure S5: A comparison of a P∼ and adaptive MPC ∼ controller using $I_1$ and $I_2$ to control the concentration of y2(Akt). (A) The response of y2 (Akt) to a Proportional controller ∼ and an adaptive linear MPC controller ∼. (B, C) Inputs $I_1$ and $I_2$, respectively, used in the P simulation. =Proportional gains: $Kp_{I_1} = Kp_{I_2} = 0.001$. (D, E) Inputs $I_1$ and $I_2$ used in the MPC simulation. Parameters: $T_s = 1$, $N = 10$, $\alpha = 0$, $\beta = 0$, $\gamma = [1, 10^5, -]$, $\theta = 0$ and $\eta = 1$.

In Figure S5 the two controllers obtain a similar performance, but the P controller ∼ offers no control on the inputs used. The P controller does not achieve robust control, as the low $EI$ is as an effect of the finely tuned gains reacting to the initial error in the output, whereas the adaptive MPC controller ∼ can be tailored for specific inputs' choices.