

Supplementary Information
Model predictive control of cancer cellular dynamics:
a new strategy for therapy design

Benjamin Smart¹, Irene de Cesare², Ludovic Renson¹, Lucia Marucci^{2,3,4}

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

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

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

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

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

S1 NSCLC Model

The model of the NSCLC is [1]

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

where the vector field $f(\cdot, \cdot)$ 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)).

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 PI3K
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	Set of three serine/threonine-specific protein kinases
x_{12}	pAkt	Active Akt
x_{13}	Akt	Active Methyl ethyl ketone
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* [1] and leads to the response shown in Figure 2.

$$\begin{aligned} \text{NSCLC Cell: } & pEGFR_0 = 800000 \mu M \quad pIGF1R_0 = 400000 \mu M \\ \text{Wild Type Cell: } & pEGFR_0 = 8000 \mu M \quad pIGF1R_0 = 800 \mu M \end{aligned} \quad (\text{S2})$$

$$\begin{aligned} SOS_0 &= 120000 \mu M & Ras_0 &= 120000 \mu M & Raf_0 &= 120000 \mu M & MEK_0 &= 600000 \mu M \\ ERK_0 &= 600000 \mu M & P90Rsk_0 &= 120000 \mu M & PI3K_0 &= 120000 \mu M & ppRaf_0 &= 120000 \mu M \\ & & PP2A_0 &= 120000 \mu M & RasGAP_0 &= 120000 \mu M & & \\ DSOS_0 &= pRas_0 = pRaf_0 = pMEK_0 = pERK_0 = pP90Rsk_0 = pPI3K_0 = 0 \mu M \end{aligned} \quad (\text{S3})$$

$$\begin{aligned}
\frac{dpEGFR}{dt} &= -k_{fEpEGFR} - k_{SOS:EpEGFR} \frac{DSOS}{KM_{SOS:E} + DSOS} - k_{PI3K:EpEGFR} \frac{PI3K}{KM_{PI3K:E} + PI3K} \quad (S4) \\
\frac{dpIGF1R}{dt} &= -k_{fIpIGF1R} - k_{SOS:IpIGF1R} \frac{DSOS}{KM_{SOS:I} + DSOS} - k_{PI3K:IpIGF1R} \frac{PI3K}{KM_{PI3K:I} + PI3K} \quad (S5) \\
\frac{dSOS}{dt} &= k_{SOS:EpEGFR} \frac{DSOS}{KM_{SOS:E} + DSOS} + k_{SOS:IpIGF1R} \frac{DSOS}{KM_{SOS:I} + DSOS} - k_{DSOS:P90pP90Rsk} \frac{SOS}{KM_{DSOS:P90+SOS}} - k_{Ras:SOS} \frac{Ras}{KM_{Ras:SOS+Ras}} \quad (S6) \\
\frac{dDSOS}{dt} &= k_{DSOS:pP90pP90Rsk} \frac{SOS}{KM_{DSOS:pP90+SOS}} - k_{SOS:EpEGFR} \frac{DSOS}{KM_{SOS:E} + DSOS} - k_{SOS:IpIGF1R} \frac{DSOS}{KM_{SOS:I} + DSOS} \quad (S7) \\
\frac{dRas}{dt} &= k_{Ras:SOS} \frac{Ras}{KM_{Ras:SOS+Ras}} - k_{Ras:Gab} \frac{pRas}{KM_{Ras:Gab+pRas}} - k_{PI3K:Ras} \frac{PI3K}{KM_{PI3K:Ras+PI3K}} - k_{Raf:Ras} \frac{Raf}{KM_{Raf:Ras+Raf}} \quad (S8) \\
\frac{dRas}{dt} &= k_{Ras:SOS} \frac{Ras}{KM_{Ras:SOS+Ras}} + k_{Ras:Gab} \frac{pRas}{KM_{Ras:Gab+pRas}} \quad (S9) \\
\frac{dRaf}{dt} &= k_{Raf:Ras} \frac{Raf}{KM_{Raf:Ras+Raf}} - k_{Raf:ppRaf} \frac{pRaf}{KM_{Raf:ppRaf+pRaf}} - k_{Raf:Akt} \frac{pAkt}{KM_{Raf:Akt+pAkt}} - k_{Raf:MEK} \frac{MEK}{KM_{Raf:MEK+MEK}} \quad (S10) \\
\frac{dRaf}{dt} &= -k_{Raf:Ras} \frac{Raf}{KM_{Raf:Ras+Raf}} + k_{Raf:ppRaf} \frac{pRaf}{KM_{Raf:ppRaf+pRaf}} + k_{Raf:Akt} \frac{pAkt}{KM_{Raf:Akt+pAkt}} \frac{pRaf}{pRaf} \quad (S11) \\
\frac{dRaf}{dt} &= -k_{Raf:Ras} \frac{Raf}{KM_{Raf:Ras+Raf}} + k_{Raf:ppRaf} \frac{pRaf}{KM_{Raf:ppRaf+pRaf}} + k_{Raf:Akt} \frac{pAkt}{KM_{Raf:Akt+pAkt}} \frac{pRaf}{pRaf} \quad (S12) \\
\frac{dMEK}{dt} &= -k_{MEK:Raf} \frac{MEK}{KM_{MEK:Raf+MEK}} + k_{MEK:P2A} \frac{pMEK}{KM_{MEK:P2A+pMEK}} - kon_3 \frac{MEK}{KM_3+I_3} \quad (S13) \\
\frac{dMEK}{dt} &= k_{MEK:Raf} \frac{MEK}{KM_{MEK:Raf+MEK}} - k_{MEK:P2A} \frac{pMEK}{KM_{MEK:P2A+pMEK}} - k_{ERK:MEK} \frac{ERK}{KM_{ERK:MEK+ERK}} - kon_3 \frac{pMEK}{KM_3+I_3} \quad (S14) \\
\frac{dPERK}{dt} &= k_{ERK:MEK} \frac{ERK}{KM_{ERK:MEK+ERK}} - k_{ERK:PP2A} \frac{pERK}{KM_{ERK:PP2A+pERK}} - k_{DSOS:ERK} \frac{ERK}{KM_{DSOS:ERK+SOS}} \quad (S15) \\
&\quad - k_{E:ERK} \frac{ERK}{KM_{E:ERK+pEGFR}} \\
\frac{dERK}{dt} &= -k_{ERK:MEK} \frac{ERK}{KM_{ERK:MEK+ERK}} + k_{ERK:P2A} \frac{pERK}{KM_{ERK:P2A+pERK}} \quad (S16) \\
\frac{dP90Rsk}{dt} &= k_{pP90:ERK} \frac{P90Rsk}{KM_{pP90:ERK+P90Rsk}} - k_{DSOS:pP90pP90Rsk} \frac{SOS}{KM_{DSOS:pP90+SOS}} \quad (S17) \\
\frac{dP90Rsk}{dt} &= kd_p P90Rsk - k_{pP90:ERK} \frac{P90Rsk}{KM_{pP90:ERK+P90Rsk}} \quad (S18)
\end{aligned}$$

$$\begin{aligned} \frac{dp_{PI3K}}{dt} = & k_{PI3K:Ras} p_{Ras} \frac{PI3K}{KM_{PI3K:Ras} + PI3K} + k_{PI3K:IP1GF1R} \frac{PI3K}{KM_{PI3K:I} + PI3K} + k_{PI3K:EP1EGFR} \frac{PI3K}{KM_{PI3K:E} + PI3K} - k_{Akt:PI3K} Akt \frac{p_{PI3K}}{KM_{Akt:PI3K} + p_{PI3K}} \\ & - kd_{PI3K} p_{PI3K} - kon_1 p_{PI3K} \frac{I_1}{K_{m_1} + I_1} \end{aligned} \quad (S19)$$

$$\frac{dPI3K}{dt} = -k_{PI3K:Ras} p_{Ras} \frac{PI3K}{KM_{PI3K:Ras} + PI3K} - k_{PI3K:IP1GF1R} \frac{PI3K}{KM_{PI3K:I} + PI3K} - k_{PI3K:EP1EGFR} \frac{PI3K}{KM_{PI3K:E} + PI3K} + kd_{PI3K} p_{PI3K} - kon_1 p_{PI3K} \frac{I_1}{K_{m_1} + I_1} \quad (S20)$$

$$\frac{dp_{Akt}}{dt} = -k_{Raf:Akt} p_{Akt} \frac{p_{Raf}}{KM_{Raf:Akt} + p_{Raf}} + k_{Akt:PI3K} Akt \frac{p_{PI3K}}{KM_{Akt:PI3K} + p_{PI3K}} - kon_2 p_{Akt} \frac{I_2}{K_{m_2} + I_2} \quad (S21)$$

$$\frac{dAkt}{dt} = -k_{Akt:PI3K} Akt \frac{p_{PI3K}}{KM_{Akt:PI3K} + p_{PI3K}} + kd_{Akt} p_{Akt} - kon_2 Akt \frac{I_2}{K_{m_2} + I_2} \quad (S22)$$

$$\frac{dRasGAP}{dt} = 0 \quad \frac{dpp_{Raf}}{dt} = 0 \quad \frac{dPP2A}{dt} = 0 \quad (S23)$$

S1.2 Conservation Equations

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

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

S1.3 Parameters in the NSCLC Model

Variable	Definition	Value, [1]
k_{fE}	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	$184912.0 \mu M$
k_{fI}	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	$161197.0 \text{ min}^{-1}$
$KM_{DSOS:P90}$	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	$0.884096 \text{ min}^{-1}$
$KM_{Raf:Ras}$	Michaelis–Menten constant for Raf deactivation by Ras	$62464.6 \mu M$
$k_{Raf:ppRaf}$	Catalytic constant for Raf deactivation by RafPP	$0.126329 \text{ min}^{-1}$
$KM_{Raf:ppRaf}$	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	$0.0213697 \text{ min}^{-1}$
$KM_{P90:ERK}$	Michaelis–Menten constant for p90Rsk activation by Erk	$763523.0 \mu M$
$k_{Akt:PI3K}$	Catalytic constant for Akt activation by PIK3	$0.0566279 \text{ min}^{-1}$
$KM_{Akt:PI3K}$	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

$$\frac{dT_{\text{target}}}{dt} = -T_{\text{target}} \frac{KonI}{Km + I}, \quad (\text{S25})$$

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 [2], I_2 -Oridonin [3] and I_3 -Pimasetib [4].

Drug	$Km(\mu M)$	$Kon(\text{min}^{-1})$	Target Molecule
I_1 - 3MA	60 [5]	0.1	PI3K
I_2 - Oridonin	8.9 [6]	0.01	Akt
I_3 - Pimasetib	2 [7]	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 (Figure S1A) or higher (Figure S1C and E) than those obtained applying feedback control; also, in all step input simulations, the Dose Index is higher.

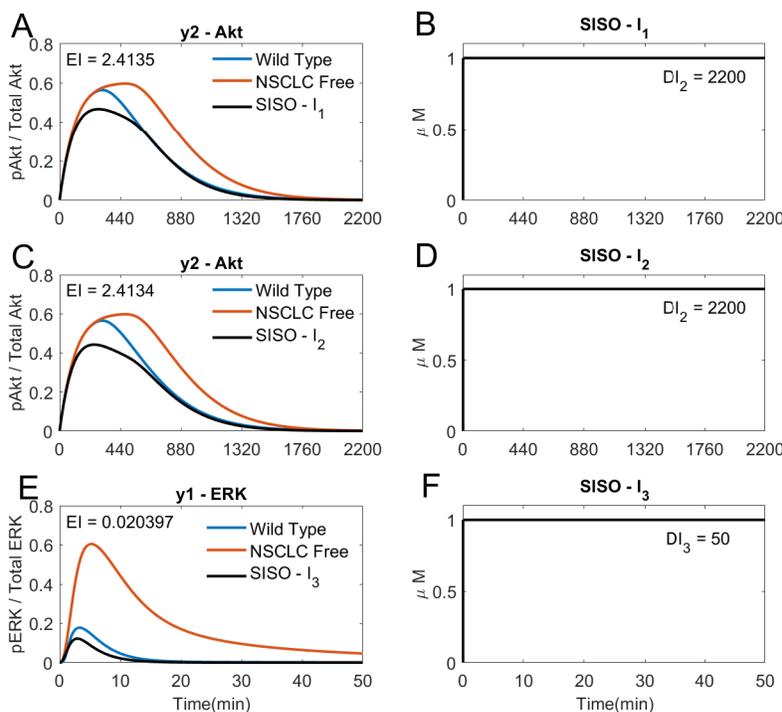


Figure S1: Step Input simulations. (A) The response of y2 (Akt) to the step input of I_1 in (B). (C) The response of y2 (Akt) to the step input of I_2 in (D). (E) The response of y1 (ERK) to the step input of I_3 in (F).

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 [8], 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.

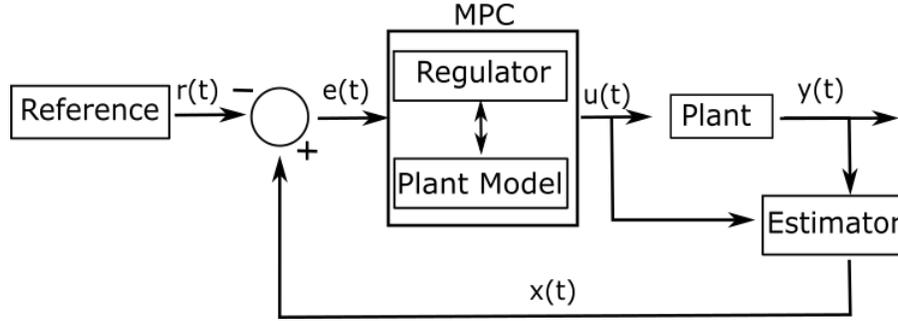


Figure S2: MPC architecture.

For the present numerical study, the plant is the non-linear 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.

$$\min_{\mathbf{U}} (J(\mathbf{E}, \mathbf{U})) \quad \text{st.} \quad -\mathbf{U} \leq \mathbf{U}_L, \quad \mathbf{U} \leq \mathbf{U}_U, \quad (\text{S26})$$

$$\text{where} \quad J(\mathbf{E}, \mathbf{U}) = \mathbf{E}^T \mathbf{Q} \mathbf{E} + \mathbf{U}^T \mathbf{R} \mathbf{U}$$

The model that the regulator sees is a discrete approximations of the NSCLC model for $1 \leq k \leq N$ steps. $\mathbf{E} = [\mathbf{e}(0), \mathbf{e}(1), \dots, \mathbf{e}(N)]^T$ describes the current and future predicted state errors. $\mathbf{U} = [\mathbf{u}(1), \mathbf{u}(2), \dots, \mathbf{u}(N)]^T$ 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 [8]. 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:

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

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

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

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

$$\mathbf{E} = \mathbf{M}\mathbf{e}_0 + \tilde{\mathbf{C}}\mathbf{U} \quad \mathbf{U} = \begin{bmatrix} \mathbf{u}(1) \\ \mathbf{u}(2) \\ \vdots \\ \mathbf{u}(N) \end{bmatrix} \quad \mathbf{M} = \begin{bmatrix} \mathbf{I} \\ \mathbf{A} \\ \vdots \\ \mathbf{A}^N \end{bmatrix} \quad \tilde{\mathbf{C}} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{B} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{AB} & \mathbf{B} & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{A}^{N-1}\mathbf{B} & \mathbf{A}^{N-2}\mathbf{B} & \dots & \mathbf{B} \end{bmatrix} \quad (\text{S29})$$

Therefore, the cost function can be rearranged as

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

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,

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

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, γ ; the error in all of the estimated states, α and the error in the outputs, β .

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

$$\mathbf{R} = \begin{bmatrix} \gamma \mathbf{I} & \dots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \dots & \gamma \mathbf{I} \end{bmatrix}$$
(S32)

where \mathbf{I} is an identity matrix. The cost function can be tuned by α , β and γ .

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

$$\left. \frac{d\mathbf{U}}{dt} \right|_k \approx \frac{\mathbf{U}(k) - \mathbf{U}(k-1)}{T_s}$$
(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.

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

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

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

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

$$\begin{aligned} J(\mathbf{U}) &= \mathbf{U}^T \mathbf{R} \mathbf{U} + (\mathbf{U}^T \tilde{\mathbf{Q}} \mathbf{U} + \tilde{\mathbf{q}} \mathbf{U}) + (\mathbf{U}^T \mathbf{D} \mathbf{U} + \mathbf{d} \mathbf{U}) \\ &= \mathbf{U}^T (\tilde{\mathbf{Q}}(\alpha, \beta) + \mathbf{R}(\gamma) + \mathbf{D}(\theta)) \mathbf{U} + (\tilde{\mathbf{q}}(\alpha, \beta) + \mathbf{d}(\theta)) \mathbf{U} \end{aligned}$$
(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 [9]. 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.

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

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

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

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.

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

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

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

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 [10]. 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 [11–13].

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:

$$EI = \int_0^T \mathbf{e}^T \mathbf{C} \mathbf{e} dt \quad (\text{S43})$$

The largest EI (worst performance) found in any MPC simulation using the chosen cost function to control y_2 (Akt) is the SISO simulation using I_1 ($EI = 2.75$, Figure 4). This has been used to normalise EI to form $\hat{E}I$.

$$\hat{E}I = \frac{EI}{\max_{\gamma_2/\gamma_1}(EI)} = \frac{EI}{EI_{I_1 \text{SISO}}} \quad (\text{S44})$$

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

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

DI_i is normalised by the DI_i of I_i 's SISO simulation (I_1 and I_2 acting on y_2 (Akt) and I_3 acting on y_1 (ERK) in Figure 4, $DI_{1 \text{SISO}} = 1068$, $DI_{2 \text{SISO}} = 546$, $DI_{3 \text{SISO}} = 4$), producing $\hat{D}I_i$.

$$\hat{D}I_i = \frac{DI_i}{DI_{i \text{SISO}}} \quad (\text{S46})$$

$\hat{D}I_i$ does not give a quantitative measure on the dose of the combined input profile. Within current literature, there are many methods of trying to summarise the joint effect and toxicity of combination therapies, where multiple drugs are given together at a determined time point [14]; 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, $\hat{D}I_i$.

An Isobole can be defined as $I_{so} = \hat{D}I_1 + \hat{D}I_2$ 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.

$$BI = \hat{D}I_1 + \hat{D}I_2 - \hat{D}I_1 \hat{D}I_2 \quad (\text{S47})$$

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 [15]. All three indexes are used in Figure 7 to compare multiple simulations.

S6 Drug Holidays

Within the adaptive MPC program, the user can set specific time intervals in which the controller does not give specific drugs or can specify that only one drug can be used at a time, in order to avoid toxicity induced by long exposure.

A discrete MISO simulation, as described in Section 3.4, where $T_s = 1 \text{min}$ can be seen in Figure S4. Figure S4 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.

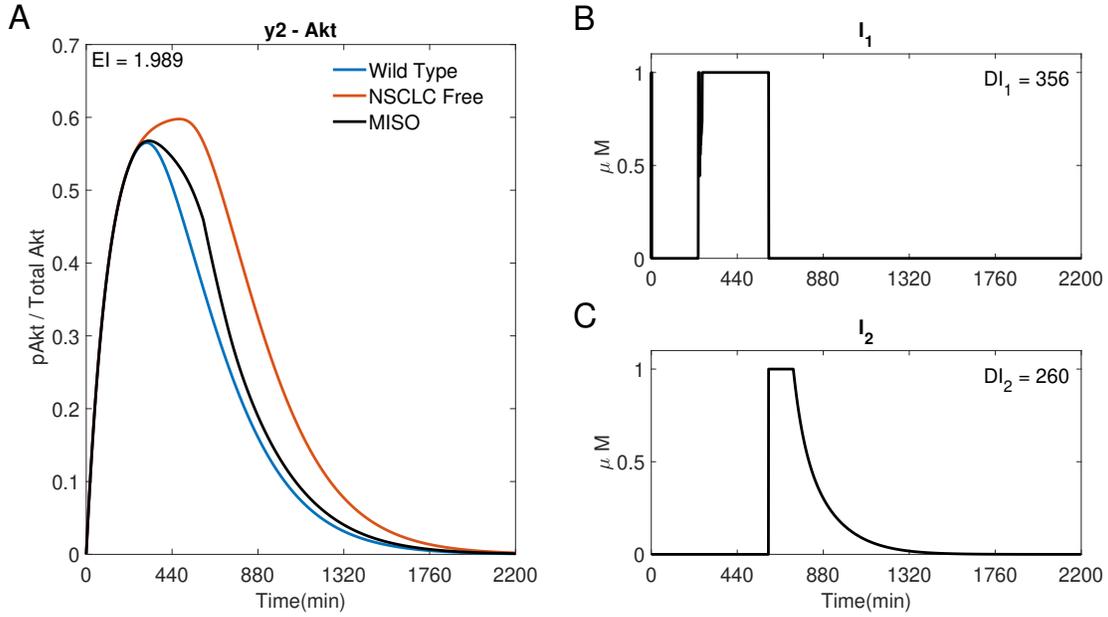


Figure S3: A MISO adaptive MPC simulation swapping I_1 and I_2 to control y_2 (Akt). (A) The response of Akt. (B, C) The inputs used in the simulation. Parameters: $T_s = 1min$, $N = 10$, $\alpha = 0$, $\beta = 0$, $\gamma = [1, \infty, -]$ when $0 \leq t \leq 600min$ and $\gamma = [\infty, 10^5, -]$ when $600 < t \leq 2200min$, $\theta = 0$ and $\eta = 1$.

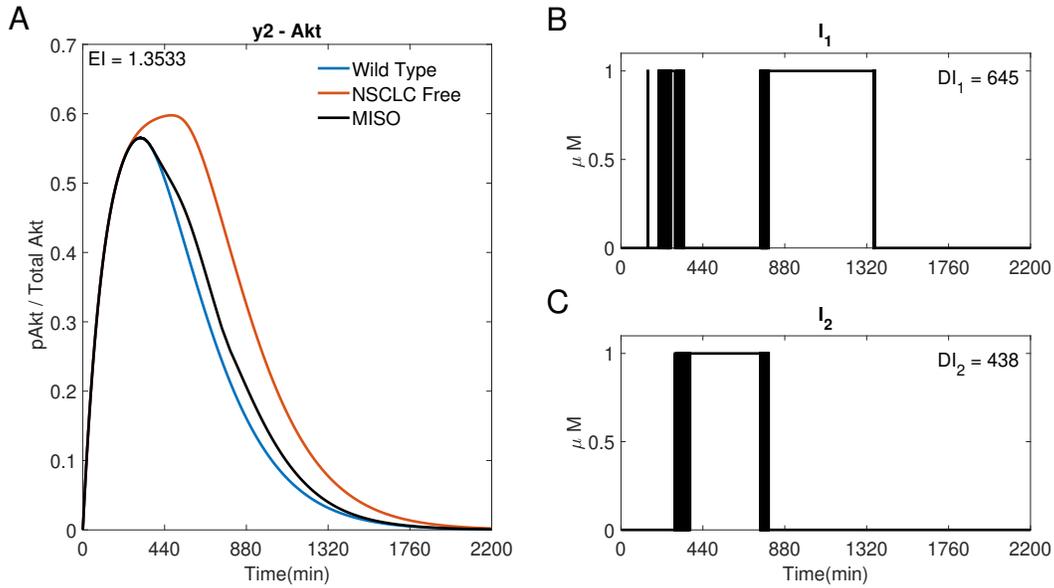


Figure S4: A discrete MISO adaptive MPC simulation using I_1 and I_2 to control the concentration of y_2 (Akt). A) The response of Akt. B) and C) The inputs used in the simulation. Parameters: $T_s = 1min$, $N = 10$, $\alpha = 0$, $\beta = 0$, $\gamma = [1, 10^5, -]$, $\theta = 0$ and $\eta = 1$.

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 with no guarantee of reaching the global minimum. The non-linear simulations have used MATLAB's 'fmincon', a gradient based non-linear solver. It is the

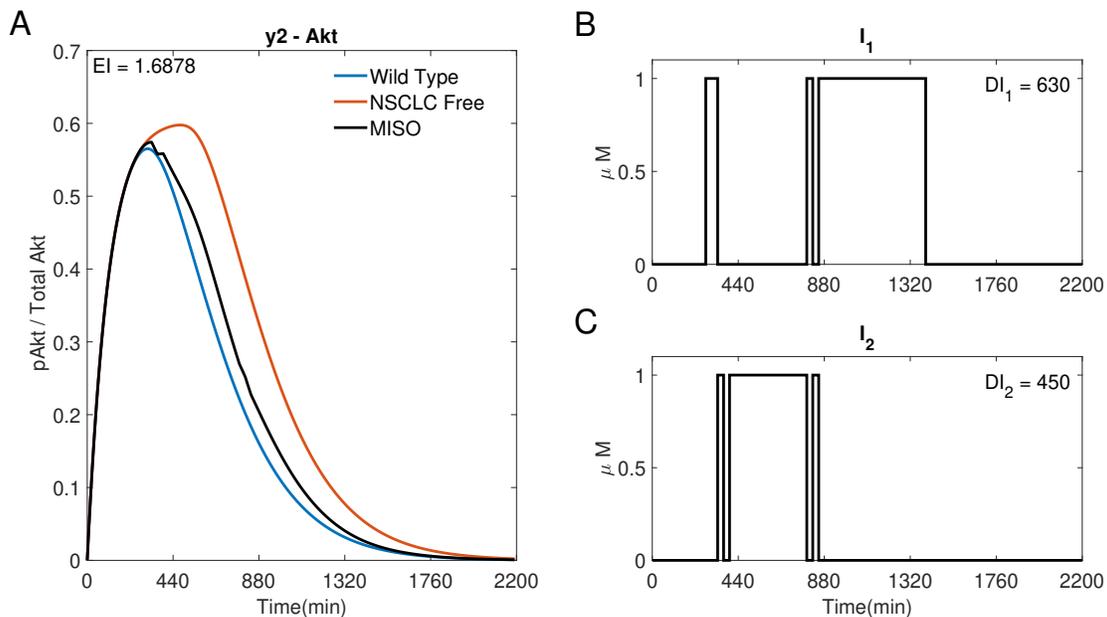


Figure S5: A discrete MISO adaptive MPC simulation using I_1 and I_2 to control y_2 (Akt). (A) The response of Akt to the inputs in (B) and (C). Parameters: $T_s = 30min$, $N = 10$, $\alpha = 0$, $\beta = 0$, $\gamma = [1, 10^5, -]$, $\theta = 0$ and $\eta = 1$.

fastest appropriate solver in MATLAB R2021b.

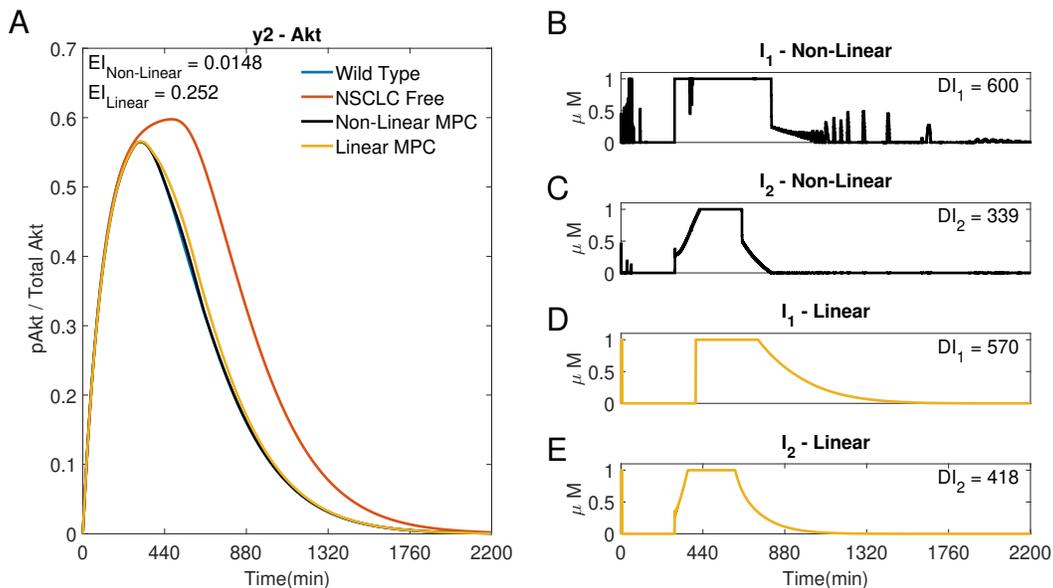


Figure S6: Two MISO MPC simulations using I_1 and I_2 to control the concentration of y_2 (Akt). (A) The response of Akt to a non-linear MPC controller (—) and an adaptive linear MPC controller (—). (B, C) Inputs I_1 and I_2 used in the non-linear simulation. (D, E) Inputs I_1 and I_2 used in the linear simulation. Parameters: $T_s = 1$, $N = 10$, $\alpha = 0$, $\beta = 0$, $\gamma = [1, 10^5, -]$, $\theta = 0$ and $\eta = 1$.

Figure S6 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 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. It has been shown that a linear MPC is more than sufficient for controlling gene expressions (e.g. [16]) and therefore in order to use a convex solver, the linear adaptive MPC has been used.

S8 MPC vs Proportional (P) Control Schemes

All feedback simulations have used an MPC controller. Figure S7 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 S7 (—), however the response is sensitive to the choice of gains. It can be seen in Figure S7 B) and C) that the inputs are identical, as the controller does not know the dynamics of the plant.

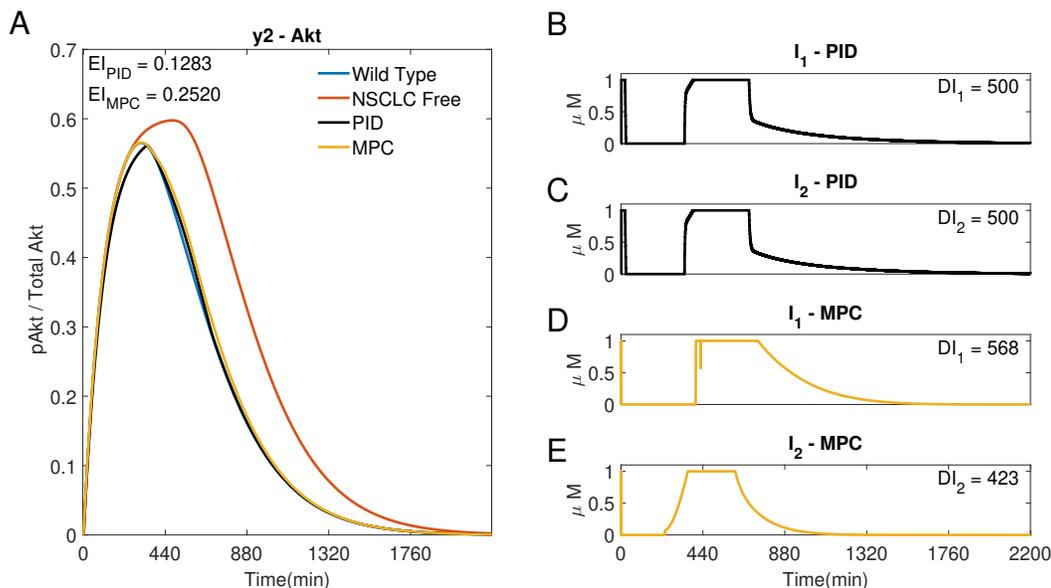


Figure S7: A comparison of a P(—) and adaptive MPC (—) controller using I_1 and I_2 to control the concentration of y2(Akt). (A) The response of y2 (Akt) to a Proportional controller (—) and an adaptive linear MPC controller (—). (B, C) Inputs I_1 and I_2 , respectively, used in the P simulation. =Proportional 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 S7 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.

S9 Identified Model with a Kalman Filter

Experimentally, the controller does not have access to noise free measurements of all the internal states. To replicate more closely experiments, we therefore made use of a Kalman filter to estimate the states that cannot be measured [17], iteratively improving the estimates of the unknown states by using an estimator gain with faster dynamics than the NSCLC system. This gain is optimally chosen to minimise the variance between the measured output and the output of the model including all the estimated states. Crucially, this estimator only has access to the output measurements and will subsequently estimate all the internal states required for the adaptive MPC program.

Thus far it has been assumed that a non-linear model of the system of interest exists. Although this is true for the NSCLC system, this is seldom true for other biological systems. Therefore, as discussed in Section 3.5, a reduced order model can be identified from input/output data and used within the Kalman filter to estimate the internal states of the reduced order model from the output and used within the MPC program to predict the future behaviour of all the internal states, but most crucially the outputs.

S9.1 Data for Model Fitting

For a given control scheme it is possible to measure various input/output data sets, which can be used to identify a model of the system. Here, the training data for model fitting consists of 91 simulated experiments sampled every minute for 1500 minutes. The first experiment is a free response. The next 30 are random series of smaller step changes of $0.2\mu M$. Starting from a random initial concentration of 0, 0.2, 0.4, 0.6, 0.8 or $1\mu M$ according to a uniform distribution, then using a uniform distribution with an 80% chance of staying at the same concentration, 10% chance of increasing by $0.2\mu M$ and a 10% chance of decreasing by $0.2\mu M$, limited to a maximum concentration of $1\mu M$ and a minimum of $0\mu M$. The inputs can change every 20 minutes. Out of the 30 experiments, 10 use only I_1 , 10 use only I_2 and 10 use both inputs. The next 30 experiments are unit steps, with a step increasing from $0\mu M$ to $1\mu M$ at a random time, initiated by a normal distribution with a mean between the reference peak and the free response peak, a standard deviation of 200 minutes and a duration according to a uniform distribution with a maximum duration of 500 minutes, after which the input steps down from $1\mu M$ to $0\mu M$. 10 experiments use only I_1 , 10 use only I_2 and 10 use both inputs. The final 30 experiments are random unit ramps, with the points where the ramp leaves and returns to the x-axis set by normal distribution, with a mean of 200 minutes and 800 minutes respectively and a standard deviation of 200 minutes. The peak of the ramp is $1\mu M$ at a time set by a uniform distribution between the two points where the input is not zero. Similarly 10 experiments use only I_1 , 10 use only I_2 and 10 use both inputs.

S9.2 Linear Model Identification

In order to use the controller in its current adaptive form (successive linearisations), the model needs to be non-linear with a known consistent structure to be linearised at each time step. Therefore a Grey box model needs to be used (as opposed to a black box), where the structure of the model is set and parameters within the model are estimated. Initially a linear Grey box model has been estimated using MATLAB's 'GREYEST' function as in [18], using the 91 data sets as discussed and the structure shown.

Due to the size of the problem, it has been decided to identify a model for $y_2(\text{Akt})$ in response to I_1 and I_2 in order to simulate this MISO response.

$$\mathbf{x} = [pEGFR; pIGF1R; pAkt]^T \quad \mathbf{u} = [I_1; I_2]^T$$

$$\begin{aligned} \dot{\mathbf{x}} &= \begin{bmatrix} p_1 & p_2 & p_3 \\ p_4 & p_5 & p_6 \\ p_7 & p_8 & p_9 \end{bmatrix} \mathbf{x} + \begin{bmatrix} p_{10} & p_{11} \\ p_{12} & p_{13} \\ p_{14} & p_{15} \end{bmatrix} \mathbf{u} \\ \mathbf{y} &= [0, 0, 1] \mathbf{x} \end{aligned} \quad (\text{S48})$$

The only constraints on the model structure are that the initial conditions of EGFR and IGF1R in the full order model are the initial conditions of states one and two in the reduced order model, and that the third state represents the output $y_2(\text{Akt})$.

The initial estimate for each of the 15 parameters was set to zero and the 'fmincon' optimiser within the 'GREYEST' function was selected. The identified parameters can be seen in Table S4. The linear model achieves a 91.24% fit to the free response and an $FPE = 3.73603 \times 10^8$ (The Akaike Final Prediction Error used within the identification toolbox, representing the optimal cost function value, scaled by the number of estimated parameters and data used).

S9.3 Non-Linear Model Identification

It has been assumed that the non-linear dynamics within the input/output data can be modelled by a quadratic couplings between the states, as discussed in [19]. Therefore the non-linear reduced order model is identified by using a 'best-case' linear model and then identifying the quadratic parameters that improve the model. MATLAB's 'NLGREYEST' function can be used with the 'lm' optimisation option to identify the non-linear grey box model as shown, all of the fixed parameters are denoted as p_i^* and are taken from the linear model in Section S9.2, whereas all the parameters to be identified are p_i , with an initial estimation of zero.

$$\dot{\mathbf{x}} = \begin{bmatrix} p_1^* \mathbf{x}(1) + p_2^* \mathbf{x}(2) + p_3^* \mathbf{x}(3) & + p_{10}^* \mathbf{u}(1) + p_{11}^* \mathbf{u}(2) & + p_{16} \mathbf{x}(1) \mathbf{x}(2) + p_{17} \mathbf{x}(2) \mathbf{x}(3) + p_{18} \mathbf{x}(3) \mathbf{x}(1) \\ p_3^* \mathbf{x}(1) + p_5^* \mathbf{x}(2) + p_6^* \mathbf{x}(3) & + p_{12}^* \mathbf{u}(1) + p_{13}^* \mathbf{u}(2) & + p_{19} \mathbf{x}(1) \mathbf{x}(2) + p_{20} \mathbf{x}(2) \mathbf{x}(3) + p_{21} \mathbf{x}(3) \mathbf{x}(1) \\ p_5^* \mathbf{x}(1) + p_8^* \mathbf{x}(2) + p_9^* \mathbf{x}(3) & + p_{14}^* \mathbf{u}(1) + p_{15}^* \mathbf{u}(2) & + p_{22} \mathbf{x}(1) \mathbf{x}(2) + p_{23} \mathbf{x}(2) \mathbf{x}(3) + p_{24} \mathbf{x}(3) \mathbf{x}(1) \end{bmatrix} \quad (\text{S49})$$

$$\mathbf{y} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \mathbf{x} = [pAkt] \quad (\text{S50})$$

Therefore it can be seen that 15 of the parameters are estimated in the linear grey box identification and the remaining 9 are estimated in the non-linear grey box identification. All identified parameter values can be seen in Table S4. The non-linear model achieves a 91.50% fit to the free response and an $FPE = 2.90675 \times 10^8$ (used within the identification toolbox as an index of fit).

S9.4 Combination Therapies

As in Section 3.3, the weight of each input within the cost function can be varied in order to produce different input profiles which achieve a similar performance. Figure S8 is plotted in the same way as Figure 7 but for the reduced order model within the adaptive MPC alongside a Kalman filter. It should be noted that the normalising values for the reduced order model are different, with $\max(EI) = EI_{I_1 SISO} = 5.538$, $DI_{I_1 SISO} = 1103$ and $DI_{I_2 SISO} = 1224$ instead of $EI_{I_1 SISO} = 2.7495$, $DI_{I_1 SISO} = 1068$ and $DI_{I_2 SISO} = 546$ from the full order model simulations, meaning that the BI of the reduced order model (Figure S8) appears lower than the BI of the full order model (Figure 7); however this is biased due to the normalisation of DI_2 by a considerably higher $DI_{I_2 SISO}$.

Linear Parameters	Value (min^{-1})	Non-Linear Parameters	Value ($min^{-1}\mu M^{-1}$)
p_1	1.84876560679288	p_{16}	3.01715352715229e-10
p_2	0.934424758794504	p_{17}	4.23335038416439e-10
p_3	-0.0552763825589960	p_{18}	-7.44239585208250e-10
p_4	-1.68009991335114	p_{19}	6.67918779037395e-10
p_5	-0.841260457140930	p_{20}	1.69580767933824e-10
p_6	0.0807205160834991	p_{21}	-3.52482688591449e-10
p_7	0.0160117099422266	p_{22}	-1.80823509525499e-10
p_8	0.0162103314558821	p_{23}	-1.65822059757745e-09
p_9	1.00272432189185	p_{24}	-1.14930484477929e-09
p_{10}	12975.8419361687		
p_{11}	25308.3253870871		
p_{12}	-24704.7367491077		
p_{13}	-47698.9440594388		
p_{14}	0		
p_{15}	0		

Table S4: Identified parameters for the linear and non-linear reduced order models.

Figure S8 shows that for the reduced order model there is a range of R which can significantly reduce both EI ($-$) and BI ($-$). Therefore, the control performance of the MISO controller is better than any SISO simulation while keeping drug concentrations low. The optimal input is associated to the minimum value of the EI ($-$) with the MISO response shown in Figure S9.

When comparing the full order model in Figure 7 to the reduced order model in Figure S8, it can be seen that there is a larger region of R in the reduced order model in which the error, EI , decreases. Using this identified model, MPC predicts that the inputs will have less of an effect on the output, when compared to the full order model. Therefore, the cost function decides to use more of each input for the same weightings. This suggests that in order to quickly identify the weights in the cost function that produce this optimum, a better model of the system would result in a sharper decline of EI to this optimum.

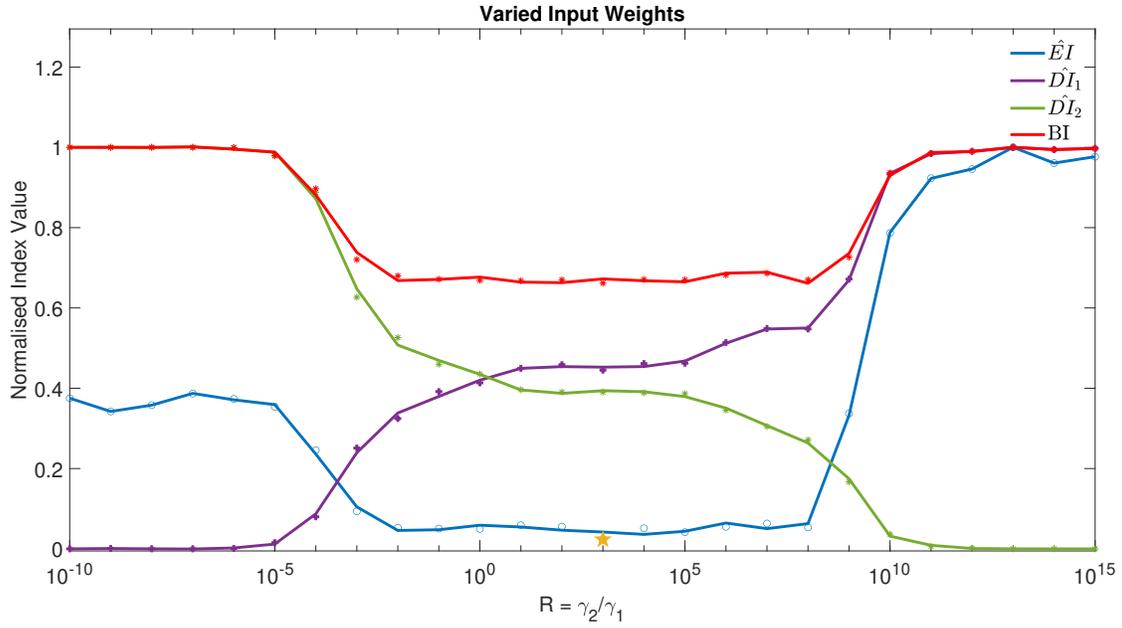


Figure S8: MISO adaptive MPC simulations based of a three state model identified from input/output data including a Kalman filter to estimate the internal states from the measured output. Using I_1 and I_2 to control y_2 (Akt) with a range of weights for the inputs. Parameters: $T_s = 1min$, $N = 10$, $\alpha = 0$, $\beta = 0$, $\theta = 0$ and $\eta = 1$.

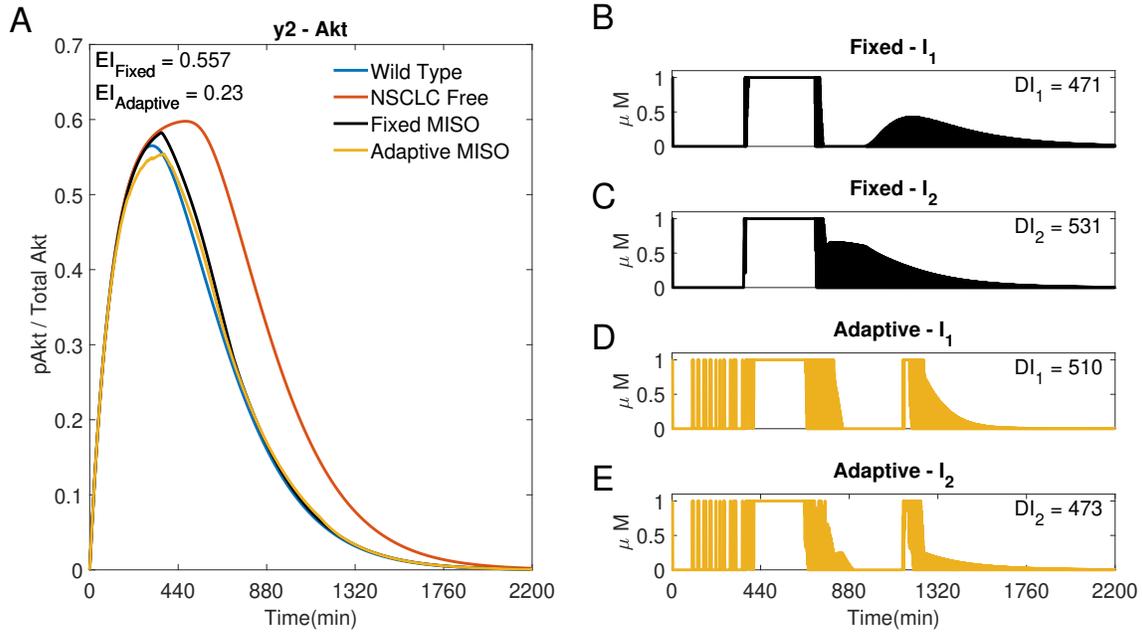


Figure S9: Two MISO adaptive MPC simulation based of a three state model identified from input/output data including a Kalman filter to estimate the internal states from the measured output, comparing the adaptive model to the linear model. Using I_1 and I_2 to control y_2 (Akt). (A) The response of Akt to the inputs in (B) and (C). Parameters: $T_s = 1min$, $N = 10$, $\alpha = 0$, $\beta = 0$, $\gamma = [1, 10^5, -]$, $\theta = 0$ and $\eta = 1$.

References

- [1] 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.
- [2] 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.
- [3] 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.
- [4] 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.
- [5] PI3K Inhibitor Review. URL: <https://www.selleckchem.com/PI3K.html>.
- [6] Akt Inhibitor Review. URL: <https://www.selleckchem.com/Akt.html>.
- [7] MEK Inhibitor Review. URL: <https://www.selleckchem.com/MEK.html>.
- [8] J. Rawlings et al. *Model predictive control : theory, computation, and design*. Nob Hill Publishing, 2nd edition, 2020.
- [9] 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.
- [10] B. Ogunnaike et al. *Process dynamics, modeling, and control*. Oxford University Press, New York and NY, 1994.
- [11] 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.
- [12] P. Mendis et al. Adaptive Model Predictive Control with Successive Linearization for Distillate Composition Control in Batch Distillation. *MERCCon 2019 - Proceedings, 5th International Multidisciplinary Moratuwa Engineering Research Conference*, pages 366–369, 7 2019. doi:10.1109/MERCCon.2019.8818777.
- [13] 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.
- [14] V. Vakil et al. Drug Combinations: Mathematical Modeling and Networking Methods. *Pharmaceutics*, 11(5), 5 2019. doi:10.3390/PHARMACEUTICS11050208.
- [15] E. Demidenkoid et al. Statistical determination of synergy based on Bliss definition of drugs independence. *PLOS ONE*, 2019. doi:10.1371/journal.pone.0224137.
- [16] I. Cesare et al. Control-Based Continuation: A New Approach to Prototype Synthetic Gene Networks. *ACS Synthetic Biology*, 6 2022. doi:10.1021/ACSSYNBIO.1C00632.
- [17] M. Grewal et al. *Kalman filtering : theory and practice with matlab*. 2014.
- [18] 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.
- [19] J. Paduart et al. Identification of nonlinear systems using Polynomial Nonlinear State Space models. *Automatica*, 46(4):647–656, 4 2010. doi:10.1016/J.AUTOMATICA.2010.01.001.