Mobile Robotics

M2-MARS

Constrained Optimization : Optimal Drug Injection for Cancer Treatment

By :
Souhaiel BEN SALEM

17 January 2022

Lab Objectives: The goal of this lab is to determine the best drug injection approach for treating cancer (reducing the population of proliferating cells) while maintaining the patient's health,because drug also affects the healthy (Quiescent) cells performance.

Introduction:

The model used to describe the effects of the drug on both the proliferating and healthy cells is the following :

$$\dot{P}(t) = [\gamma - \delta - \alpha - u(t)]P(t) + \beta Q(t)$$
$$\dot{Q}(t) = \alpha P(t) - (\lambda + \beta)Q(t)$$

Where:

- P: the number of proliferating cells
- Q: the of Quiescent (healthy) cells
- γ:the proliferating cells growth rate
- δ: the proliferating cells natural death rate
- α: the transition rate from proliferating state to the quiescent state
- β: the transition rate from quiescent to proliferating state
- λ: the cells di erentiation rate (mature bone-marrow cells leaving the bone marrow to enter blood stream)
- $u(t)$ is the drug control variable that acts on the proliferating cells.

The optimization problem we seek to solve is the following:

$$\min_{u(\cdot)} \left[P(T) + \mu \cdot \int_0^T u(\tau) d\tau \right]$$
under the constraints $u(t) \in [0, 1]$ and $P(t) + Q(t) \geq \rho$ for all $t \in [0, T]$

QUESTIONS

1. Declaring the model's coefficients as well as other variables globally in the main script for future use:

```
clear all
global gamma alpha lambda delta beta T rho nu P0 Q0
gamma = 1.46; alpha = 5.63; lambda = 0.16; delta = 0; beta = 0.48;
T = 30; rho = 0.35; nu = 0.001; P0 = 0.5; Q0 = 0.5; x0 = [0.5 0.5 0];
```

2. Creating the function $u(t) = lesu(i)$ that selects a control value u such that $t \in [(i-1)\frac{T}{N}, i\frac{T}{N}[$ from $lesu$ which is a vector of N successive control actions defining a piece-wise constant control:

```
function u = udet(t,lesu)
global T
N = length(lesu);
if t == T
        u = lesu(N);
end
for i = 1:1:N
    if t >= (i-1)*T/N && t < i*T/N
        u = lesu(i);
end
end
```

Verification:

```
>> lesu=[0.5 ;0.8 ;0.3 ;1.9 ;0.54]

lesu =

    0.5000
    0.8000
    0.3000
    1.9000
    0.5400

>> u =udet(T/3,lesu)

u =

    0.8000
```

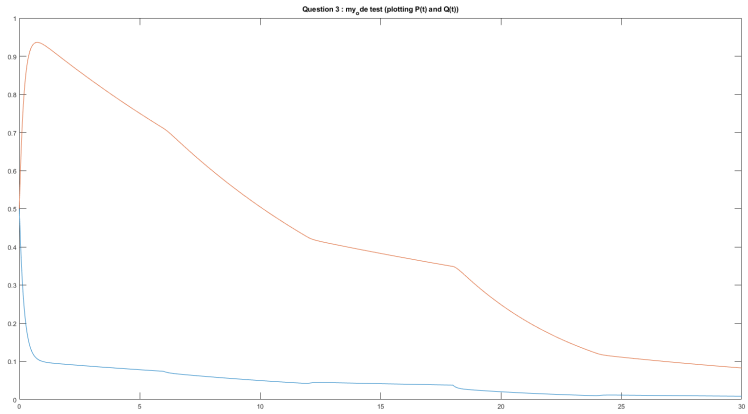
3. Writing a function that implements the ordinary differential equations governing the change in the number of Quiescent and Proliferating cells defined in the introduction. Here, the control input u is added to make use of its integration later in the cost function :

```
function xdot = my_ode(t,x,lesu)
global gamma alpha lambda delta beta
u = udet(t, lesu);
xdot = [(gamma-delta-alpha-u)*x(1) + beta*x(2); alpha*x(1) - (lambda + beta)*x(2); u];
end
```

To verify this function , we use the following code:

```
[tt,xx]=ode45(@my_ode,[0 T],[P0 Q0 0],[],lesu);
figure(1)
plot(tt,xx(:,1:2));
title('Question 3 : my_ode test (plotting P(t) and Q(t))')
```

Which yields the following figure:



4. Now we will use the above function, to write a function that implements the cost function $J = cost(lesu)$ defined in the introduction. Note that the integral of $u(t)$ over time is the third component of xx, since we defined u as the third component of $xdot$. The cost function depends on $P(t)$ and the integral of $u(t)$ between 0 and T, therefore we need extract these values from xx. We do that by taking the final value of the first and third dimensions of xx.
The cost function J is therefore:

```
function J = cost(lesu)
global gamma alpha lambda delta beta mu P0 Q0 T
[tt,xx]=ode45(@my_ode,[0 T],[P0 Q0 0],[],lesu);
J = xx(end,1) + nu*xx(end,3);
end
```

5. According to the documentation of **fmincon**, we must define the **equality** (C_{eq}) and **inequality** (C) constraints. In this case, we don't have equality constraints so we will be passing an empty vector. To guarantee that $P(t) + Q(t) > \rho$ at each instant, we can take $C = max(\rho - (P(t) + Q(t))$.

Hence, our constraints function can be written as follows:

```
function [C, Ceq] = constraints (lesu)
global P0 Q0 T rho
Ceq = [];
[~,xx]=ode45(@my_ode,[0 T],[P0 Q0 0],[],lesu);
C = max(-xx(:,1)-xx(:,2)+rho);
end
```

6. Optimal Drug injection strategies for $P(0) = Q(0) = 0.5$, $N \in 2, 5, 10$ (N N successive control actions defining a piece-wise constant control) and starting from the initial guess $lesu = (0, 0, ..., 0)$:

   - **N=2** :
     The optimal control sequence $uopt = (0.0196,\ 0.7618)^T$

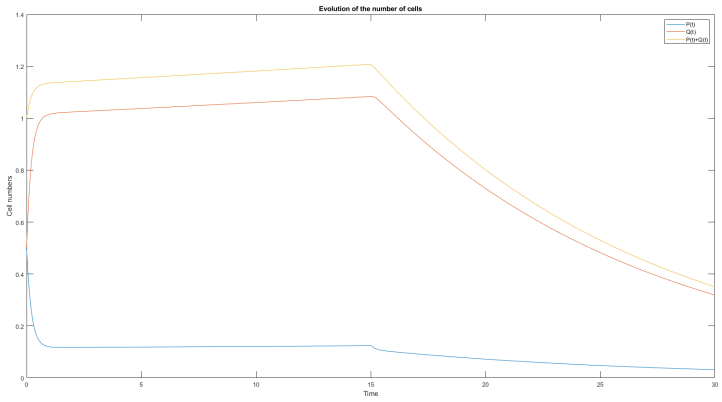


Figure 1: optimal injection strategies

   - **N=5** :
     The optimal control sequence is: $uopt = (0.3540,\ 0.1923,\ 0.1983,\ 0.3272,\ 0.6982)^T$
     We can plot both the optimal control sequence u and optimal injection strategies as follow:

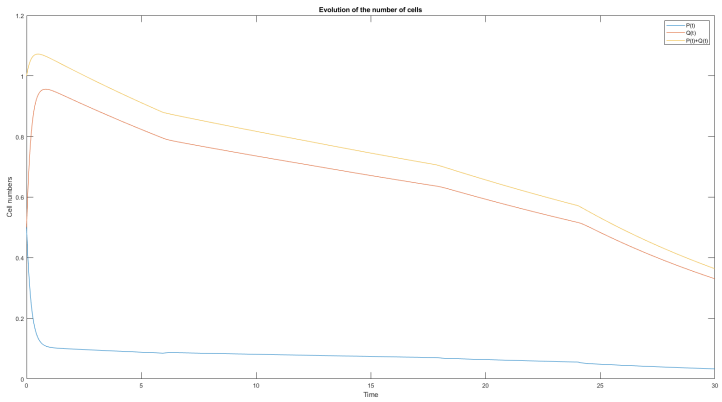


Figure 2: optimal injection strategies

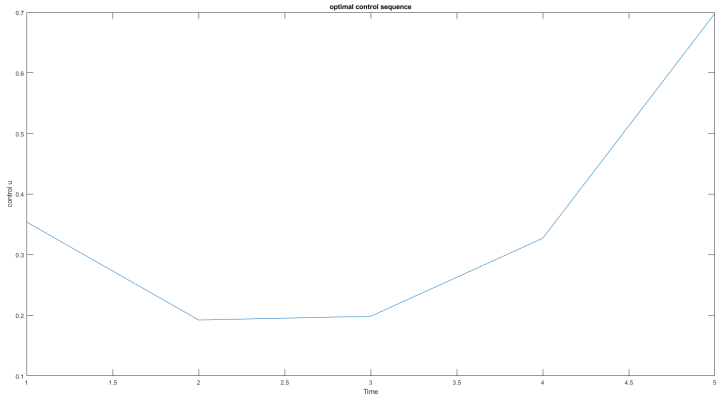


Figure 3: optimal control sequence

- **N=10** :
  The optimal control sequence is:
  $uopt = (0.3749,\ 0.4788,\ 0.2879,\ 0.1669,\ 0.2572,\ 0.3265,\ 0.3672,\ 0.1508,\ 0.2682,\ 0.9999)^T$
  We can plot both the optimal control sequence u and optimal injection strategies as follow:

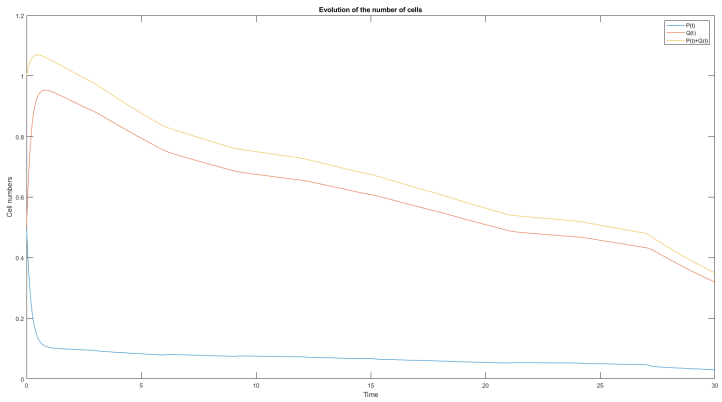


Figure 4: optimal injection strategies

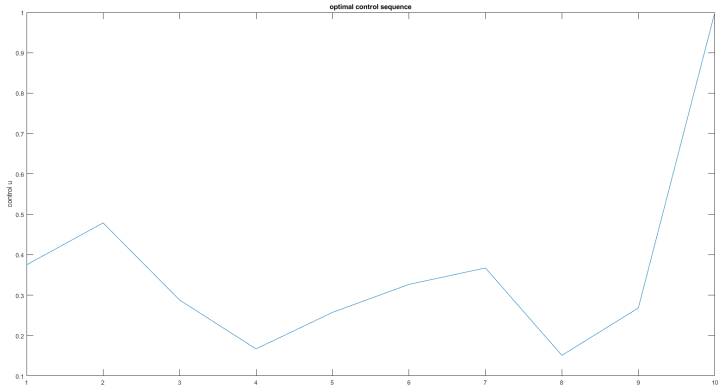


Figure 5: optimal control sequence

$\Rightarrow$ We notice that the more control actions we take, the smoother the evolution of number of cells is.

7. Repeating the previous question usingrandomly generated initial guesses for $lesu$:

   - **N=2** :
     The optimal control sequence $uopt = (0.0063,\ 0.7794)^T$ We can plot both the optimal control sequence u and optimal injection strategies as follow:

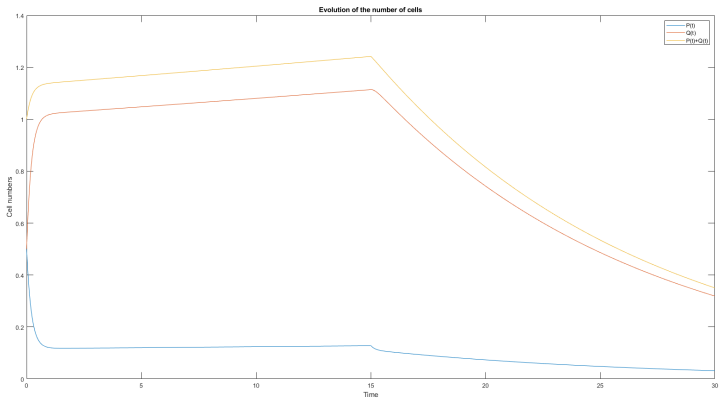


Figure 6: optimal injection strategies

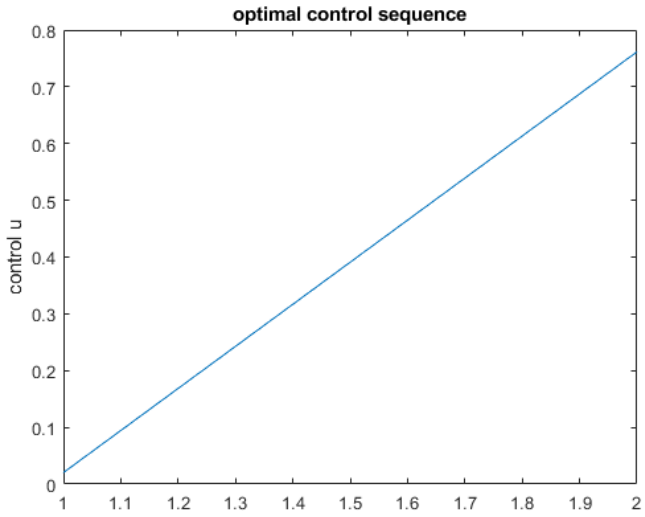


Figure 7: optimal control sequence

- **N=5** :
  The optimal control sequence $uopt = (0.3884,\ 0.1954,\ 0.1600,\ 0.1463,\ 0.9886)^T$
  We can plot both the optimal control sequence u and optimal injection strategies as follow:

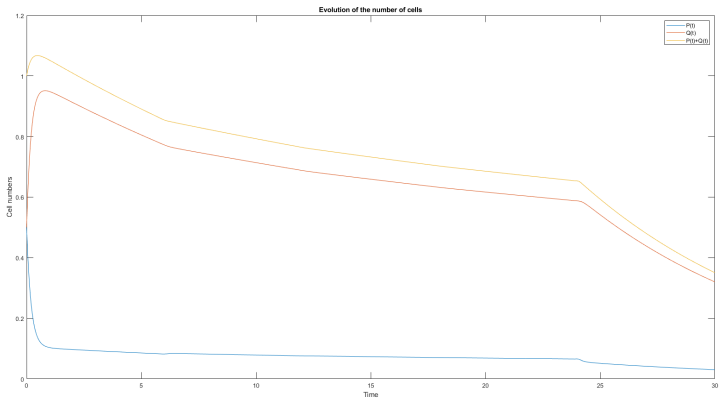


Figure 8: optimal injection strategies

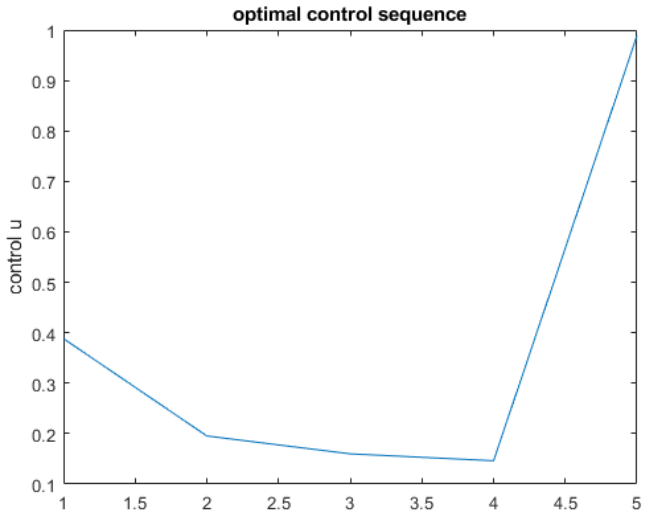


Figure 9: optimal control sequence

- **N=10** :
  The optimal control sequence is:
  $uopt = (0.2671,\ 0.3395,\ 0.3846,\ 0.2598,\ 0.1946,\ 0.1774,\ 0.2254,\ 0.3251,\ 0.5511,\ 0.9560)^T$
  We can plot both the optimal control sequence u and optimal injection strategies as follow:

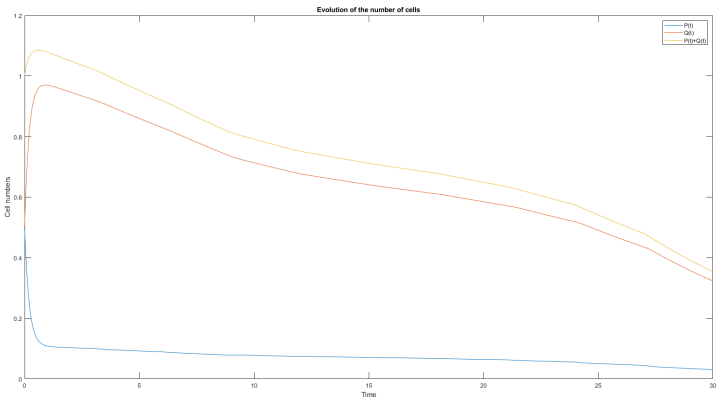


Figure 10: optimal injection strategies

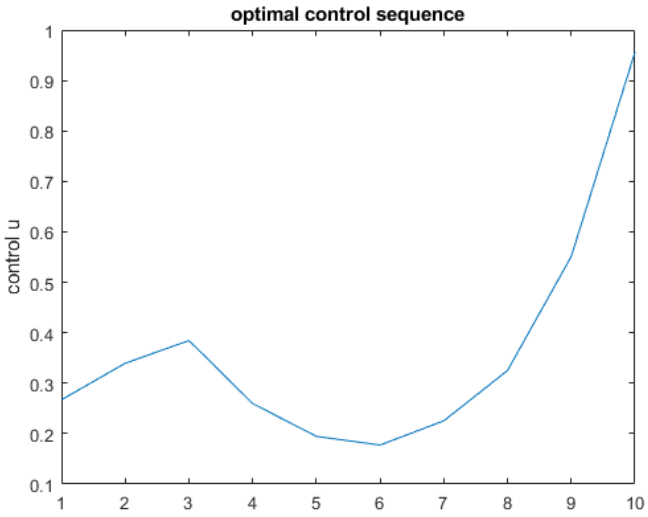


Figure 11: optimal control sequence

⇒ We notice that even though the initial control sequence is different each time we run the main function, the final , optimal solutions are almost equal i.e whatever the initial condition, fmincon always trie to converge to the best, optimal solution.

8. Using $\mu = 0.01$ and $\mu = 0.1$ for $N = 10$ and $lesu = [0, 0...0]$:
   - $\mu = 0.01$:

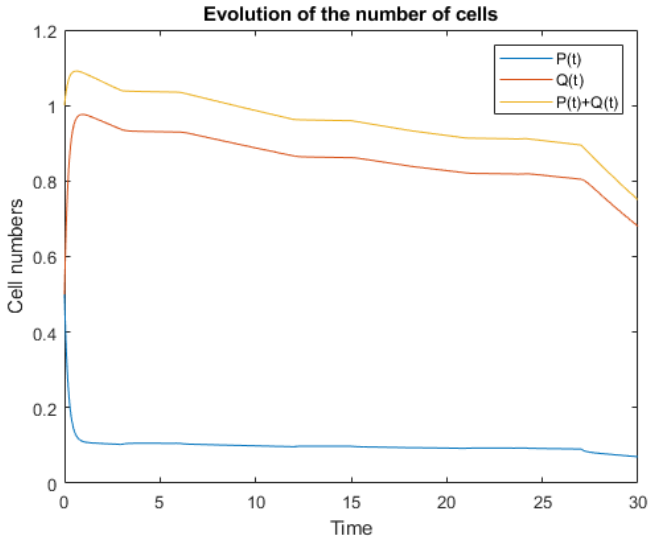


Figure 12: optimal injection strategies

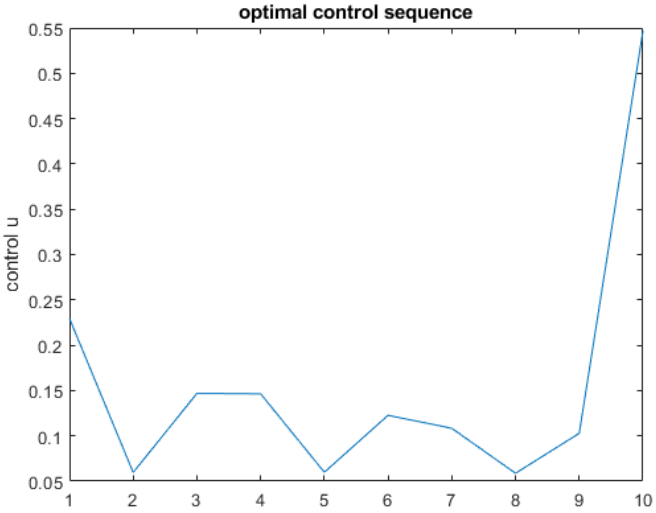


Figure 13: optimal control sequence

- $\mu = 0.1$:

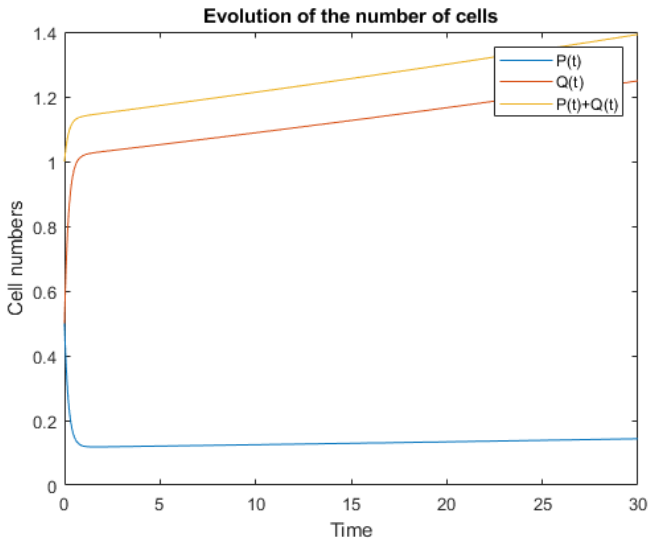


Figure 14: optimal injection strategies

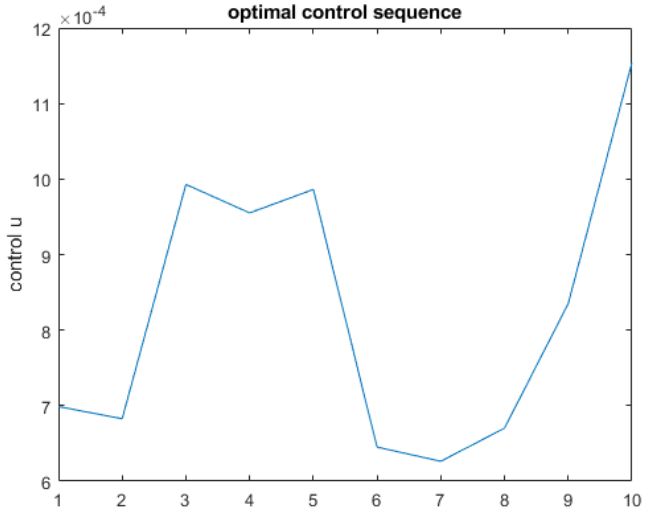


Figure 15: optimal control sequence

9. **Observations**:
   We notice that when increasing the weight of u (increasing μ) in the cost J, more healthy cells survive. However, when the weight on u is too high, the injection fails to kill the proliferating cells

as well. In a sense, we are caring so much for the patient's healthy cells that the doses we are giving him not do not kill the proliferating cells effectively (u is now too small). In fact, the control sequence we are getting in this case in not optimal for the patient's health.
Globally, we notice that by optimizing the medication injection technique, this method might assist lower treatment costs by reducing the number of replicating cells while protecting the patient's health. As a result, we can strike a good balance between efficiency and efficacy.
A solution should try to divide the injection period into more parts because this way the bad cells will be diminished gradually, avoiding shock for the body.

Conclusion:

During this lab, we tried to optimize the injection strategy for cancer treatment under constraints using MATLAB optimization tools such as the fmincon function and ode45 ODE solver. This allowed us to better understand nonlinear optimization and its real-world application.