Modelação matemática e controlo ótimo

Cristiana J. Silva

CIDMA - Centro de Investigação e Desenvolvimento em Matemática e Aplicações,
Departamento de Matemática, Universidade de Aveiro
cjoaosilva@ua.pt

Laboratórios de Computação e Visualização Científica
Módulo 4 - 2021

Optimal control applied to epidemiological models

The physical processes which take place in technology are, as rule, **controllable**, i.e., they can be realized by various means depending on the will of the man.

? question ?

How to find the very best (in one sense or another) or the **optimal control** of the process.

Examples:

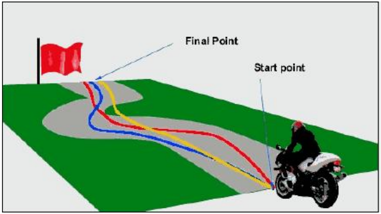
Achieve the aim of the process in

the **shortest time**;

with a **minimum expenditure of energy**.

A **control system** is a dynamic system on which one can act by using a command or control.

Optimal control theory analyzes the properties of this system with the **purpose of bringing it from an initial state to a certain final status**, and possibly respecting some constraints.



Control system

$$\frac{\dot{x}_i}{dt} = f^i(x_1, ..., x_n, u_1, ..., u_m)\,, \quad i = 1, ..., n\,; \tag{1}$$

In order to determine the course of the control system (1) in a certain time interval $t_0 \leq t \leq t_1$, it is sufficient to give the control parameters $u_1, ..., u_m$:

$$u_j = u_j(t)\,, \quad j = 1, ..., m\,; \tag{2}$$

as functions of time on this time interval.

boundary conditions

Let us assume that there exists a control $u_j = u_j(t)\,,\quad j = 1, ..., m$ which transfers the controlled object from a given initial phase state $x_i(t_0) = x_i^0\,,\quad i = 1, ..., n$ to a prescribed terminal phase state

$$x_i(t_1) = x_i^1\,,\quad i = 1, ..., n\,,$$

Request!

It is required to find a control

$$\overline{u}_t(t)\,, \quad j = 1, ..., m\,,$$

that transfers the controlled object from the initial to the final state in such a manner that the functional

$$J = \int_{t_0}^{t_1} f^0(x_1, ..., x_n, u_1, ..., u_m)\, dt\,,$$

has a **minimum value**.

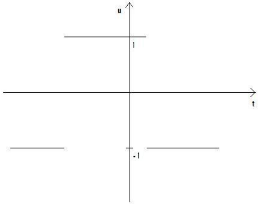
Restriction on the controls

Example: determine the position of a machine's controllers

Here the control parameters $u_1, \ldots, u_m$ cannot assume arbitrary values, but **are subject to certain restrictions**.

Because of the argument of the mechanism described by system (1), the parameter u_1 can, let us say, only assume values which satisfy the condition

$$|u_1| \leq 1\,.$$



Control region

$(u_1, ..., u_m) \in U$, where U is a subset of the space coordinates $u_1, ..., u_m$.
The choice of the set U reflects specific features of the object (1).
In the mathematical statement of the problems the set U (the "control region") is considered as arbitrary, but in technical problems the case where U is a closed set is particularly important and characteristic.

Consider

(i) the time-optimal problem for system (1);

(ii) the right-hand sides are linear functions of the variables $x_1, \dots, x_n, u_1, \dots, u_m$ with constant coefficients;

(iii) the set U is a closed convex polyhedron.

For example, U may be the *cube* defined by the inequalities:

$$|u_j| \leq 1\,, \quad j = 1, \dots, m\,.$$

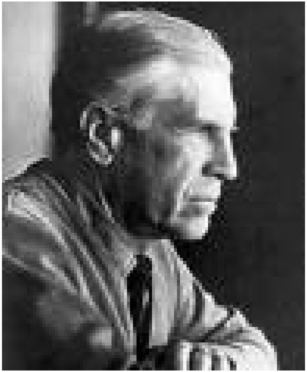
In this case it turns out that the optimal control is realized by the point $(u_1(t), \dots, u_m(t))$ which is located, in turns, at various vertices of U. The rules according to which the control jumps from one vertex to another also give the optimal control law. **Classical methods are completely unapplicable to solve such problem.**

Under these assumptions the necessary conditions for optimality are formulated in the form of a **maximum principle**:

Pontryagin Maximum Principle

Lev Semenovich Pontryagin

(3 September 1908 - 3 May 1988)
He was born in Moscow.
His **maximum principle** is fundamental to the Optimal Control theory.

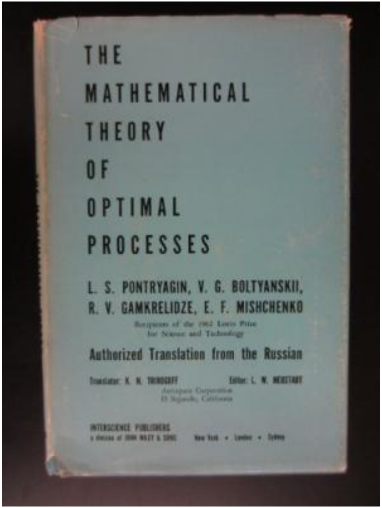


Under these assumptions the necessary conditions for optimality are formulated in the form of a **maximum principle**:

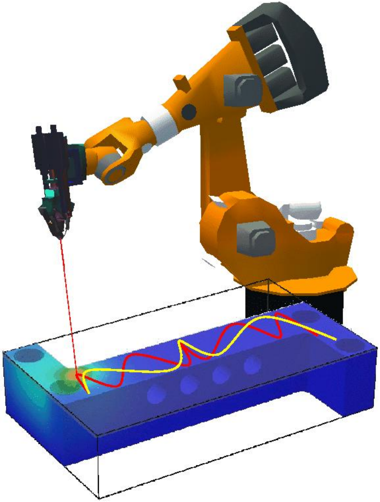
Pontryagin Maximum Principle

Lev Semenovich Pontryagin

(3 September 1908 - 3 May 1988)
He was born in Moscow.
His **maximum principle** is fundamental to the Optimal Control theory.



The control systems origins are multiple and very diverse:



- mechanical
- electricity
- electronic
- biology
- chemistry
- economy
- epidemiological
- ...

HIV/AIDS model

Let's come back to the HIV/AIDS model and rewrite it in the following way:

$$\begin{cases} S'(t) = bN(t) - \lambda(t)S(t) - \mu S(t) \\ I'(t) = \lambda(t)S(t) - (\rho + \phi + \mu)I(t) + \alpha A(t) + \omega C(t) \\ C'(t) = \phi I(t) - (\omega + \mu)C(t) \\ A'(t) = \rho\, I(t) - (\alpha + \mu + d)A(t)\,. \end{cases} \tag{3}$$

with

$$\lambda(t) = \frac{\beta}{N(t)}\left(I(t) + \eta_C\, C(t) + \eta_A A(t)\right),$$

We have just rewrote the recruitment rate Λ as $\Lambda = bN(t)$ where b represents the inflow in the susceptible individuals.

HIV/AIDS model - normalized model

In the situation where the total population size $N(t)$ is not constant, it is often convenient to consider the proportions of each compartment of individuals in the population, namely

$$s = S/N, \quad i = I/N, \quad c = C/N, \quad r = R/N\,.$$

The state variables s, i, c and a satisfy the following system of differential equations:

$$\begin{cases} s'(t) = b(1 - s(t)) - \beta(i(t) + \eta_C c(t) + \eta_A a(t))s(t) + d\, a(t)\, s(t) \\ i'(t) = \beta\,(i(t) + \eta_C\, c(t) + \eta_A a(t))\, s(t) - (\rho + \phi + b)i(t) + \alpha a(t) + \omega c(t) \\ c'(t) = \phi i(t) - (\omega + b)c(t) + d\, a(t)\, c(t) \\ a'(t) = \rho\, i(t) - (\alpha + b + d)a(t) + d\, a^2(t) \end{cases} \tag{4}$$

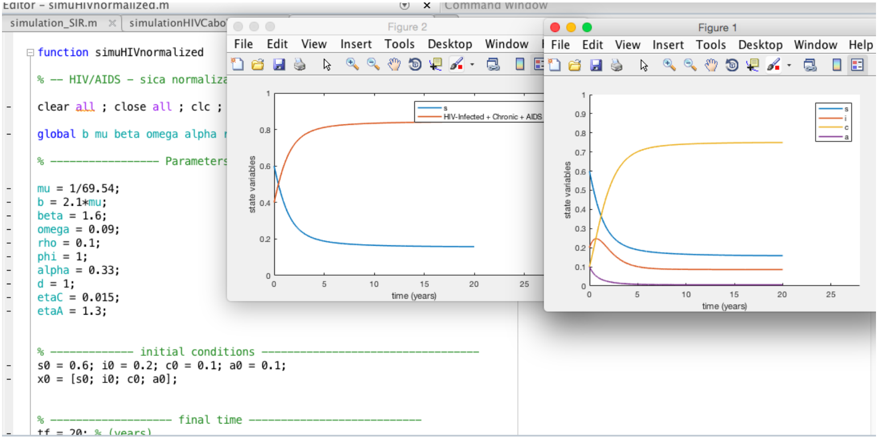
with $s(t) + i(t) + c(t) + a(t) = 1$, for all $t \in [0, T]$.

HIV/AIDS model - normalized model - Matlab

Matlab file: simuHIVnormalized.m

Parameter values: $\mu = \frac{1}{69.54}$; $b = 2.1\mu$; $\beta = 1.6$; $\eta_C = 0.015$; $\eta_A = 1.3$; $\phi = 1.0$; $\rho = 0.1$; $\alpha = 0.33$; $\omega = 0.09$; $d = 1.0$.

Initial conditions: $s(0) = 0.6$; $i(0) = 0.2$; $c(0) = 0.1$; $a(0) = 0.1$.



HIV/AIDS model - normalized model - Matlab

For the solution of the normalized HIV/AIDS model using the direct methods:

- Euler's Method;
- Runge–Kutta of Order Two;
- Runge–Kutta of Order Four;

see

C. Campos, C. J. Silva, D. F. M. Torres, *Numerical Optimal Control of HIV Transmission in Octave/MATLAB*, Math. Comput. Appl. 25 (2020), no. 1, 20 pp.

Doi: https://www.doi.org/10.3390/mca25010001
https://arxiv.org/abs/1912.09510

Optimal Control of HIV Transmission

Introduce a control function $u(\cdot)$ in model (4), which represents the effort on HIV prevention measures, such as condom use (used consistently and correctly during every sex act) or oral pre-exposure prophylasis (PrEP).
Control system:

$$
\begin{cases}
s'(t) = b(1 - s(t)) - (1 - u(t))\beta(i(t) + \eta_C c(t) + \eta_A a(t))s(t) + d\, a(t)\, s(t) \\
i'(t) = (1 - u(t))\beta\,(i(t) + \eta_C\, c(t) + \eta_A a(t))\, s(t) - (\rho + \phi + b)i(t) + \alpha a(t) + \omega c \\
c'(t) = \phi i(t) - (\omega + b)c(t) + d\, a(t)\, c(t) \\
a'(t) = \rho\, i(t) - (\alpha + b + d)a(t) + d\, a^2(t).
\end{cases}
\tag{5}
$$

The control $u(\cdot)$ is bounded: $0 \leq u \leq u_{\max}$, with $u_{\max} < 1$.
When $u = 0$, no extra preventive measure for HIV transmission is being used by susceptible individuals.
Assume that u_{max} is never equal to 1 (more realistic).

Optimal Control of HIV Transmission

Goal: find the optimal value u^* of the control u, along time, such that the associated state trajectories s^*, i^*, c^*, and a^* are solution of the system (5) in the time interval $[0, T]$ with initial given conditions

$$s(0) \geq 0\,, \quad i(0) \geq 0\,, \quad c(0) \geq 0\,, \quad a(0) \geq 0\,, \tag{6}$$

and $u^*(\cdot)$ MAXIMIZES the objective functional given by

$$J(u(\cdot)) = \int_0^T \left(s(t) - i(t) - u^2(t)\right)\, dt, \tag{7}$$

which considers the fraction of susceptible individuals (s) and HIV infected individuals without AIDS symptoms (i), and the cost associated with the support of HIV transmission measures (u).

Optimal Control of HIV Transmission - numerical solution using AMPL and NEOS SOLVER - IPOPT

We will use a direct method that consist in the discretization of the optimal control problem, reducing it to a nonlinear programming problem.

We will write the discretized optimal control problem using the AMPL language.

https://ampl.com/

Optimal Control of HIV Transmission - numerical solution using AMPL and NEOS SOLVER - IPOPT

To solve the discretized problem using the AMPL language we save the file as a **.mod** file.

See AMPL file: OptimalControlHIV.mod

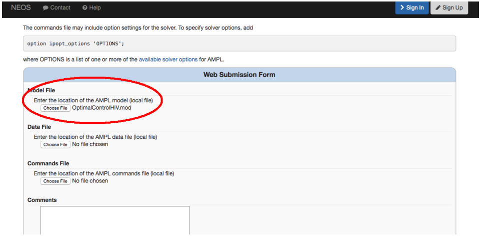
```
#####################################################
#                                                   #
#        Optimal control of HIV transmission        #
#           Normalized  SICA model                  #
#                                                   #
#####################################################

 param tf   := 20;
 param n    := 2000;
 param h    := tf/n;

 param umax = 0.5 ;

 ## initial values of state variables

 param S0 := 0.6 ;
 param I0 := 0.2 ;
 param C0 := 0.1 ;
 param A0 := 0.1 ;

##  parameters

 param mu := 1/69.54;
 param b := 2.1*mu;
 param beta := 1.9;
 param omega := 0.09;
 param rho := 0.1;
 param phi := 1;
 param alpha := 0.33;
 param d := 1;
 param etaC := 0.015;
 param etaA := 1.3;
```

```
##  state variables

 var int {i in 0..n};
 s.t. iv_int : int[0] = 0 ;

 var S {i in 0..n};
 s.t. iv_s : S[0] = S0 ;

 var I {i in 0..n};
 s.t. iv_I : I[0] = I0 ;

 var C {i in 0..n};
 s.t. iv_C : C[0] = C0 ;

 var A {i in 0..n};
 s.t. iv_A : A[0] = A0 ;

## control variables

 var u {i in 0..n} := umax  ;
 s.t. mu_c {i in 0..n} : 0 <= u[i] <= umax  ;
```

```
## right hand sides of ODEs

 var fint {i in 0..n} = S[i] - I[i] - u[i]*u[i];

 var fS {i in 0..n} = b - b*S[i] - (1 - u[i])*beta*(I[i] + etaC*C[i] +
 etaA*A[i])*S[i] + d*A[i]*S[i];

 var fI {i in 0..n} = (1-u[i])*beta*(I[i] + etaC*C[i] + etaA*A[i])*S[i] -
 (rho + phi + b)*I[i] + alpha*A[i]  + omega*C[i] + d*A[i]*I[i]  ;

 var fC {i in 0..n} = phi*I[i] - (omega + b)*C[i] +  d*A[i]*C[i] ;

 var fA {i in 0..n} = rho*I[i] - (alpha + b + d)*A[i] + d*A[i]*A[i] ;

##  objective functional

 maximize OBJ :  int[n] ;

## EULER method for ODEs

 s.t. l_int {i in 0..n-1} : int[i+1] = int[i] + h * fint[i] ;

 ## Implicit EULER method for ODEs

 s.t. l_S {i in 0..n-1}  : S[i+1] = S[i] + 0.5*h * (fS[i]+fS[i+1])  ;
 s.t. l_I {i in 0..n-1}  : I[i+1] = I[i] + 0.5*h * (fI[i]+fI[i+1])   ;
 s.t. l_C {i in 0..n-1}  : C[i+1] = C[i] + 0.5*h * (fC[i]+fC[i+1])  ;
 s.t. l_A {i in 0..n-1}  : A[i+1] = A[i] + 0.5*h * (fA[i]+fA[i+1])  ;
```

Optimal Control of HIV Transmission - numerical solution using AMPL and NEOS SOLVER - IPOPT

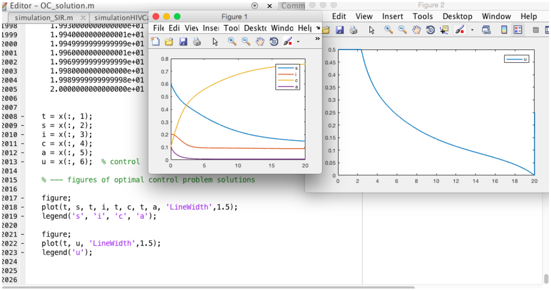
1 - go to the following web page
https://neos-server.org/neos/solvers/nco:Ipopt/AMPL.html
2 - upload the *OptimalControlHIV.mod* file in



3 - don't forget to: write your email to receive the results!

Optimal Control of HIV Transmission - make figures of the results from IPOPT using MATLAB

Select the data from NEOS IPOPT that you receive in your email:
6 columns: one for time, one for each state variable s, i, c, a and one for the control u.
Copy and then paste in MATLAB.
See Matlab file: *OC-solution.m*

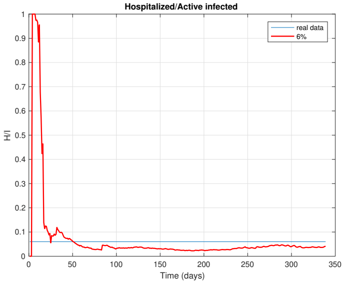


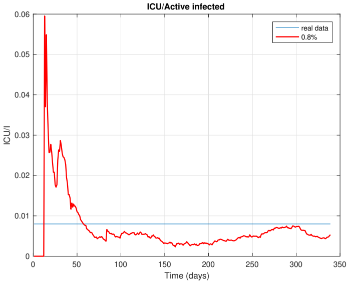
Optimal Control of HIV Transmission - make figures of the results from IPOPT using MATLAB

Don't forget to interpret the results!

Optimal Control applied to COVID-19 model

Hospitals and intensive care units occupancy beds by COVID-19 (until February 3, 2021)





Official real data, from March 02, 2020 to February 3, 2021, for the fraction of hospitalized individuals and in ICU due to COVID-19, with respect to the active infected individuals.

Optimal control: introduction of the control and its optimization

Challenges:

- reopening of the economy while preserving the health of the population without collapsing the public health system;
- keep the schools open (children under 10 years old are not obliged to use a mask in Portugal) and prevent the economy to sink;
- there is a minimum number of people that need to be susceptible to infection;
- need to account that the population do not always follow the rules imposed by governments;

Goals:

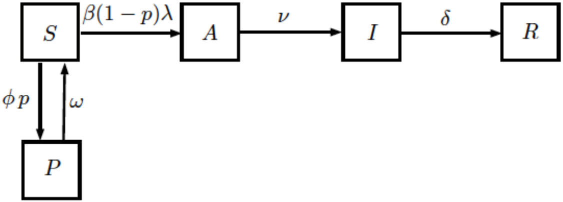
- Develop tools to quantify this effect and include it into the equations.
- Investigate the use of optimal control theory to design strategies for this phase of the disease.

Optimal control problem: main goal

Goal:

- maximize the number of people transferred from class P to the class S (that helps keeping the economy alive) and, simultaneously, minimize the number of active infected individuals and, consequently, the number of hospitalized and people needing ICU (in other words, ensuring that the health system is never overloaded);
- impose that the number of active infected cases is always below 60% of the maximum value observed up to July 29, 2020 ($I_{\max}$). This condition warrants that the health system does not collapse.

SAIRP model: constant population

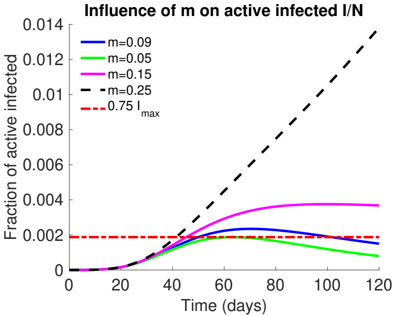


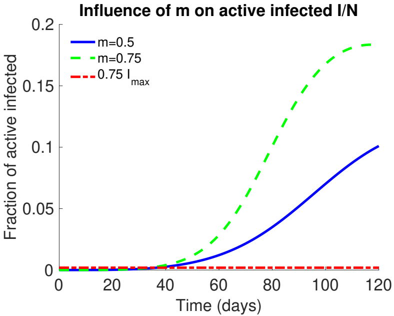
$$
\begin{cases}
\dot{S}(t) = -\beta(1-p)\left(\theta A(t) + I(t)\right) S(t) - \phi p S(t) + \omega P(t), \\
\dot{A}(t) = \beta(1-p)\left(\theta A(t) + I(t)\right) S(t) - \nu A(t), \\
\dot{I}(t) = \nu A(t) - \delta I(t), \\
\dot{R}(t) = \delta I(t), \\
\dot{P}(t) = \phi p S(t) - \omega P(t).
\end{cases}
$$

S, A, I, R, P, **represent fractions of the population:**
$S + A + I + R + P = 1$.

Sensitivity of class I with respect to parameter m

- Parameter m in model SAIRP: represents the fraction of protected individuals P that is transferred to susceptible S;
- the class of active infected individuals I is very sensitive to the change of the parameter m.





The dotted red line marks the level $0.75 \times I_{\max}$ that represents approximately 75% of the maximum fraction of active infected cases observed in Portugal (up to July 29, 2020).

Consider the fixed parameters $(\beta, p) = (1.464, 0.675)$ and all the other parameters from previous table.

Optimal control problem: control system

The parameter m in the *SAIRP* model, is replaced by a control function $u(\cdot)$.
The control $u(\cdot)$ represents the fraction of individuals in class P of *protected* that is transferred to the class S.

Control system:

$$\begin{cases} \dot{S}(t) = -\beta(1-p)\left(\theta A(t) + I(t)\right) S(t) - \phi p S(t) + w u(t) P(t), \\ \dot{A}(t) = \beta(1-p)\left(\theta A(t) + I(t)\right) S(t) - \nu A(t), \\ \dot{I}(t) = \nu A(t) - \delta I(t), \\ \dot{P}(t) = \phi p S(t) - w u(t) P(t). \end{cases} \tag{8}$$

Control constraints: $0 \leq u(t) \leq u_{\max}$ with $u_{\max} \leq 1$. In other words, the solutions of the problem must belong to the following set of admissible control functions:

$$\Theta = \left\{ u,\ u \in L^1\left([0, t_f], \mathbb{R}\right) \mid 0 \leq u(t) \leq u_{\max} \quad \forall\ t \in [0, t_f] \right\}. \tag{9}$$

Optimal control problem: cost functional and state constraint

Mathematically, the main goal consists to **minimize the cost functional**

$$J(u) = \int_0^{t_f} k_1 I(t) - k_2\, u(t)\, dt\,, \tag{10}$$

representing the fact that we want to minimize the fraction of infected individuals I and, simultaneously, maximize the intensity of letting people from class P go back to class S. The constants k_i, $i = 1, 2$, represent the weights associated to the class I and control u.

State constraint: Moreover, the solutions of the optimal control problem must satisfy:
$I(t) \leq \zeta$ **with** $\zeta = 0.6 \times I_{\max}$.

Optimal control problem: numerical simulations

For the numerical simulations, we considered:

- $k_1 = 100$, $k_2 = 1$;
- $t_f = 120$ days;
- $(\beta, \delta) = (1.464, 1/30)$, $m = 0.09$, $p = 0.675$;
- all the other parameters from previous table.

Numerically, we:

- discretized the optimal control problem to a nonlinear programming problem, using the Applied Modeling Programming Language (AMPL);
- after, the AMPL problem was linked to the optimization solver IPOPT;
- the discretization was performed with $n = 1500$ grid points using the trapezoidal rule as the integration method.

Optimal control problem: AMPL file

See AMPL file: `OCcovidPT_SAIRPstateconstraint.mod`

```
######################################################
#                                                    #
#        Optimal control - covidPT - SAIRP           #
#                                                    #
#                                                    #
######################################################

 param tf   := 120;
 param n    := 1200 ;
 param h    := tf/n;

 param u1max := 0.5 ; # podemos considerar outros valores 0 < umax < 1

 param K1 = 100;
 param K2 = 1;

 ## initial values of state variables

 param N0 := 10295909;

 ## cada classe representa a fracao em relacao a populacao total

 param S0 := 10295894/N0;
 param A0 := (2/0.15)/N0;
 param I0 := 2/N0 ;
 param P0 := 0/N0 ;
```

```
##  parameters

 param beta := 1.464;
 param theta := 1;
 param p := 0.675;
 param phi := 1/12;

 param v := 1/1;
 param q := 0.15;
 param nu := v*q;

 param delta := 1/27;

 param m := 0.09;
 param w := 1/45;


##  state variables

 var int {i in 0..n};
 s.t. iv_int : int[0] = 0 ;

 var S {i in 0..n};
 s.t. iv_S : S[0] = S0 ;
 s.t. mu_V {i in 0..n} : S[i] >= 0 ;

 var A {i in 0..n};
 s.t. iv_A : A[0] = A0 ;
 s.t. mu_A {i in 0..n} : A[i] >= 0 ;
```

```
var I {i in 0..n};
s.t. iv_I : I[0] = I0 ;
s.t. mu_I {i in 0..n} : I[i] >= 0 ;
s.t. sc_I {i in 0..n} : I[i] <= 2/3*2.5*10^(-3) ;

var P {i in 0..n};
s.t. iv_P : P[0] = P0 ;
s.t. mu_P {i in 0..n} : P[i] >= 0 ;


# control variables

var u1 {i in 0..n}   ;
s.t. mu_c1 {i in 0..n} : 0 <= u1[i] <= u1max  ;


# right hand sides of ODEs

var fint {i in 0..n} =  K1*I[i] - K2*u1[i];
```

Optimal control problem: NEOS IPOPT solver

See AMPL file: `OCcovidPT_SAIRPstateconstraint.mod`

neos SERVER RESULTS

```
*************************************************************

  NEOS Server Version 6.0
  Job#     : 10468138
  Password : pIfcWSiB
  User     :
  Solver   : nco:Ipopt:AMPL
  Start    : 2021-05-11 06:33:46
  End      : 2021-05-11 06:33:59
  Host     : prod-sub-1.neos-server.org

  Disclaimer:

  This information is provided without any express or
  implied warranty. In particular, there is no warranty
  of any kind concerning the fitness of this
  information for any particular purpose.

  Announcements:

  We will require an email address  with all job submissions
  on the NEOS Server starting on January 8, 2021.
  Please see:
    https://neos-guide.org/content/FAQ#email

*************************************************************
You are using the solver ipopt.

%% COMMENTS %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

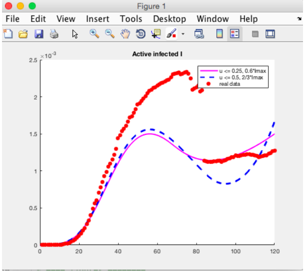
```
Number of Iterations....: 208

                                   (scaled)                 (unscaled)
Objective...............:  -1.2095994964601060e+01   -1.2095994964601060e+01
Dual infeasibility......:   2.0862671877626307e-07    2.0862671877626307e-07
Constraint violation....:   6.4028782276182028e-12    6.4028782276182028e-12
Complementarity.........:   4.2648463744261285e-09    4.2648463744261285e-09
Overall NLP error.......:   4.2648463744261285e-09    2.0862671877626307e-07


Number of objective function evaluations             = 382
Number of objective gradient evaluations             = 192
Number of equality constraint evaluations            = 382
Number of inequality constraint evaluations          = 0
Number of equality constraint Jacobian evaluations   = 224
Number of inequality constraint Jacobian evaluations = 0
Number of Lagrangian Hessian evaluations             = 208
Total CPU secs in IPOPT (w/o function evaluations)   =      4.201
Total CPU secs in NLP function evaluations           =      0.915

EXIT: Optimal Solution Found.

Ipopt 3.13.4: Optimal Solution Found

suffix ipopt_zU_out OUT;
suffix ipopt_zL_out OUT;
OBJ = -12.096
```

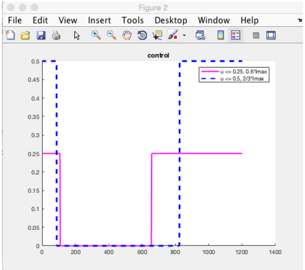
0.0000000000000000e+00	9.9999854311066660e-01	1.2950127408209741e-06	1.9425191112314610e-07	0.0000000000000000e+00	4.9999995986831397e-01
1.0000000000000001e-01	9.9439236084441829e-01	1.3475478216861288e-06	2.1331635928563683e-07	5.6061099119631011e-03	4.9999998457685357e-01
2.0000000000000001e-01	9.8882381256658403e-01	1.4023206897104663e-06	2.3311365088006374e-07	1.1174582792915430e-02	4.9999998420764408e-01
2.9999999999999999e-01	9.8329264551240347e-01	1.4594031504371113e-06	2.5367511899621451e-07	1.6705671301706249e-02	4.9999998382963423e-01
4.0000000000000002e-01	9.7779860861107504e-01	1.5188687872518565e-06	2.7503306929131227e-07	2.2199626400358170e-02	4.9999998344250834e-01
5.0000000000000000e-01	9.7234145247433179e-01	1.5807929828465259e-06	2.9722080317359666e-07	2.7656697365442615e-02	4.9999998304593485e-01
5.9999999999999998e-01	9.6692092938509377e-01	1.6452529403312451e-06	3.2027264121892524e-07	3.3077131799378601e-02	4.9999998263956541e-01
6.9999999999999996e-01	9.6153679328619834e-01	1.7123277039773319e-06	3.4422394681414130e-07	3.8461175641655430e-02	4.9999998222303432e-01
8.0000000000000004e-01	9.5618879976920423e-01	1.7820981795726492e-06	3.6911115002070163e-07	4.3809073179979764e-02	4.9999998179595700e-01
9.0000000000000002e-01	9.5087670606327157e-01	1.8546471543713959e-06	3.9497177165181437e-07	4.9121067061347556e-02	4.9999998135792906e-01
1.0000000000000000e+00	9.4560027102411681e-01	1.9300593166204865e-06	4.2184444755609069e-07	5.4397398303041439e-02	4.9999998090852504e-01
1.1000000000000001e+00	9.4035925512304175e-01	2.0084212746448373e-06	4.4976895310047589e-07	5.9638306303554010e-02	4.9999998044729682e-01
1.2000000000000000e+00	9.3515342043603589e-01	2.0898215754740914e-06	4.7878622784498735e-07	6.4844028853437430e-02	4.9999997997377232e-01
1.3000000000000000e+00	9.2998253063295344e-01	2.1743507229935316e-06	5.0893840040155648e-07	7.0014802146080080e-02	4.9999997948745378e-01
1.3999999999999999e+00	9.2484635096676115e-01	2.2621011956021643e-06	5.4026881346904162e-07	7.5150860788410406e-02	4.9999997898781595e-01
1.5000000000000000e+00	9.1974464826285995e-01	2.3531674633612274e-06	5.7282204903625734e-07	8.0252437811528840e-02	4.9999997847430433e-01
1.6000000000000001e+00	9.1467719090847854e-01	2.4476460046166375e-06	6.0664395374464522e-07	8.5319764681267882e-02	4.9999997794633300e-01
1.7000000000000000e+00	9.0964374884213761e-01	2.5456353220792038e-06	6.4178166440199995e-07	9.0353071308681282e-02	4.9999997740328228e-01
1.8000000000000000e+00	9.0464409354318565e-01	2.6472359583467406e-06	6.7828363363845305e-07	9.5352586060462277e-02	4.9999997684449649e-01
1.8999999999999999e+00	8.9967799802140502e-01	2.7525505108525314e-06	7.1619965569571775e-07	1.0031853576929177e-01	4.9999997626928117e-01
2.0000000000000000e+00	8.9474523680668827e-01	2.8616836462249718e-06	7.5558089234039877e-07	1.0525114574411684e-01	4.9999997567690019e-01
2.1000000000000001e+00	8.8984558593878404e-01	2.9747421140435542e-06	7.9647989889198249e-07	1.1015063978035983e-01	4.9999997506657246e-01
2.2000000000000002e+00	8.8497882295711250e-01	3.0918347599767562e-06	8.3895065035593506e-07	1.1501724017005881e-01	4.9999997443746880e-01
2.2999999999999998e+00	8.8014472689064849e-01	3.2130725382877899e-06	8.8304856765216354e-07	1.1985116771193961e-01	4.9999997378870775e-01
2.3999999999999999e+00	8.7534307824787472e-01	3.3385685236945695e-06	9.2883054392891832e-07	1.2465264172142004e-01	4.9999997311935163e-01
2.5000000000000000e+00	8.7057365900680062e-01	3.4684379225707027e-06	9.7635497095205424e-07	1.2942188004054672e-01	4.9999997242840205e-01
2.6000000000000001e+00	8.6583625260505137e-01	3.6027980834747323e-06	1.0256817655594111e-06	1.3415909904786488e-01	4.9999997171479466e-01

Print results

See MATLAB file: `covidPT_OC.mod`

```
function covidPT_OC

clear all; close all; clc; format long;

% ficheiro que resulta do NEOS - solucao OC
M=xlsread('data-pt-covid19-15abril');

x=load('resultado_neos_umax025state060.txt');  % umax = 0.25 e state constraint <= 0.6

x2 = load('neos_umax05_state2_3.txt'); % umax = 0.5 e state constraint <= 2/3 Imax

t = x(:, 1);
S = x(:, 2);
A = x(:, 3);
I = x(:, 4);
P = x(:, 5);
u = x(:, 6); % control

% ---- figures --------
N0 = 10295894 + 2 + 2/0.15;
time = 1:1:120;

figure;
hold on
plot(t, I,'m-', 'Linewidth', 2);
plot(x2(:, 1), x2(:, 4),'b--', 'LineWidth', 2);
plot(time, M(time, 2)./N0,'r*', 'LineWidth', 2);
title('Active infected I');
legend('u <= 0.25, 0.6*Imax', 'u <= 0.5, 2/3*Imax', 'real data');
```





Don't forget to interpret the results!