Universidad de Chile
Facultad de Ciencias Físicas y Matemáticas
Departamento de Ingeniería Matemática

Laboratorio 1

Matlab y Bocop

Integrantes:	Felipe Olivares
Profesor:	Héctor Ramírez C.
Auxiliares:	Antoine Haddon
	Emilio Molina
Ayudante:	Pablo Arratia

Fecha de entrega: 9 de octubre de 2018
Santiago, Chile

Índice de Contenidos

Lista de Figuras

Lista de Códigos

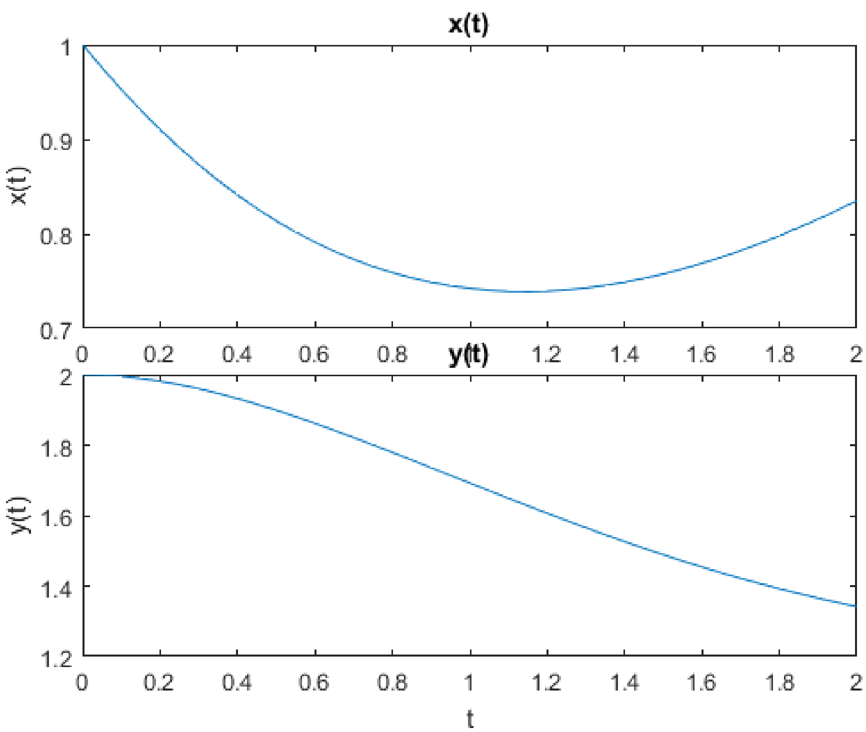
1. Matlab

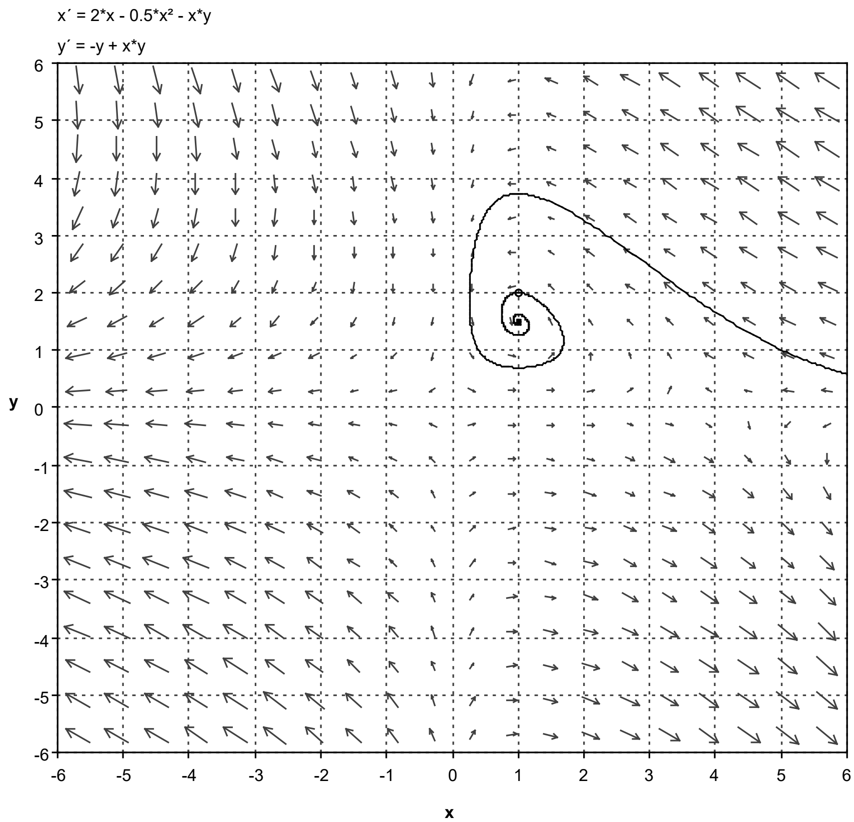
1.1. Ejercicio 1

Queremos resolver el siguientes sistema

$$\begin{cases} \dot{x} = 2x - \frac{1}{2}x^2 - xy \\ \dot{y} = -y + xy \end{cases}$$

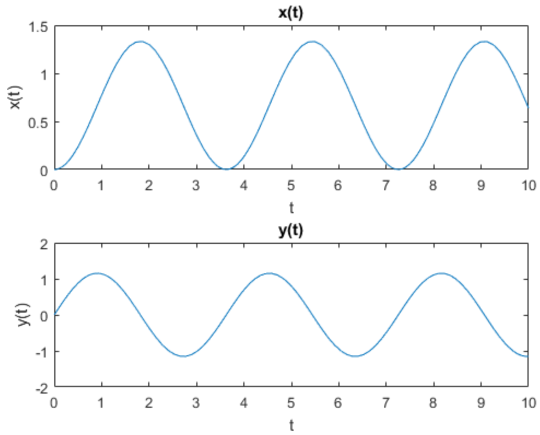
Con condiciones iniciales $x_0 = 1$ e $y_0 = 2$. Al utilizar dsolve() para resolverlo sin condiciones iniciales da que no puede encontrar una solución explícita. Para resolverlo con las condiciones iniciales se utilizó ode45 en el intervalo [0,2]. Los resultados son los siguientes:

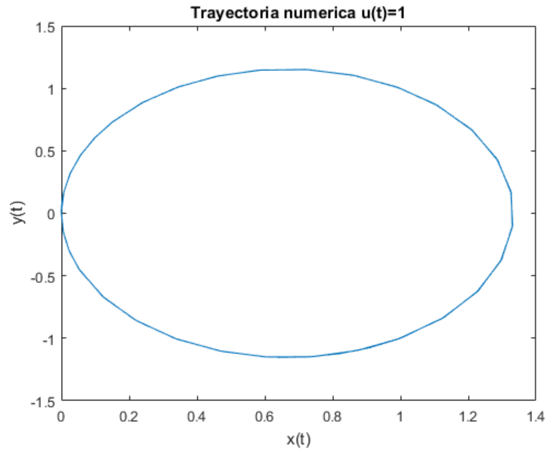


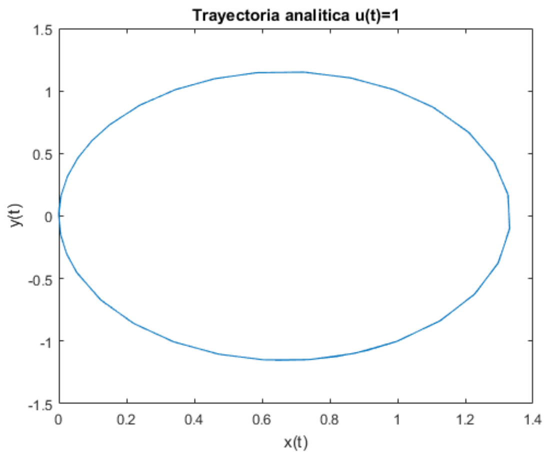


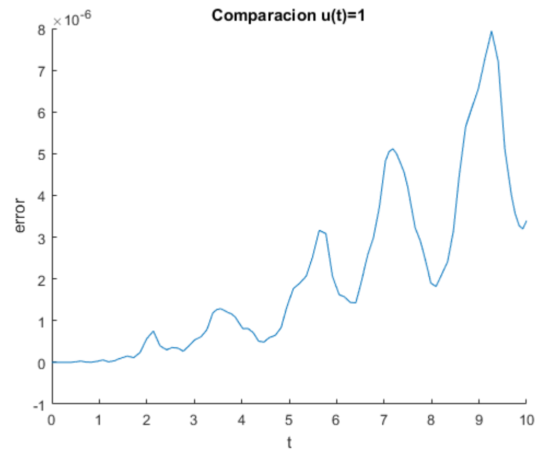
1.2. Ejercicio 2

Las soluciones usando los diferentes controles son las siguientes:
u(t) = 1:

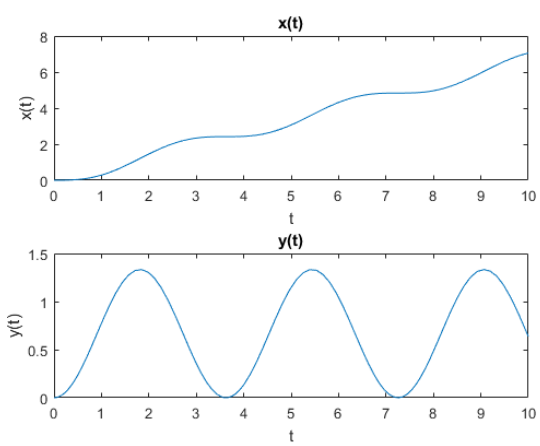

x(t)
x(t)
t
y(t)
y(t)
t

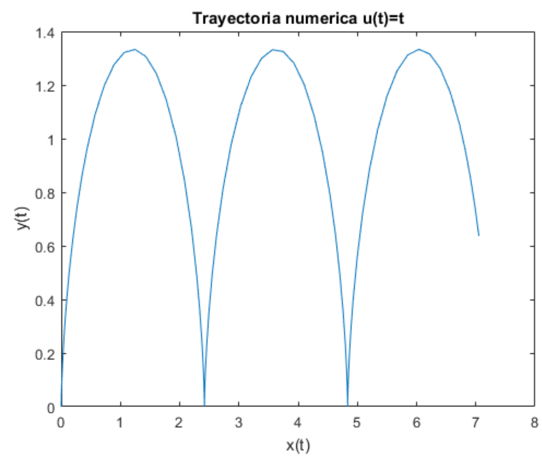

Trayectoria numerica u(t)=1
y(t)
x(t)

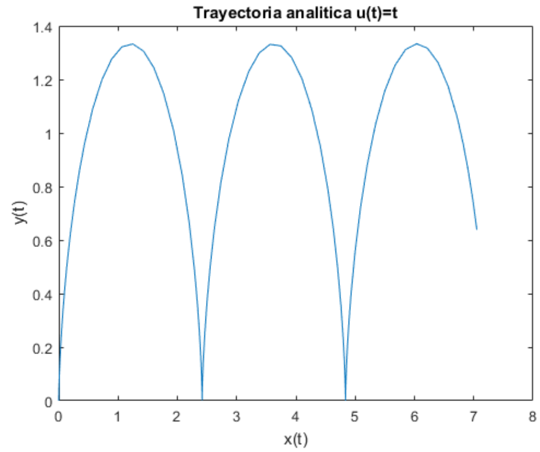

Trayectoria analitica u(t)=1
y(t)
x(t)

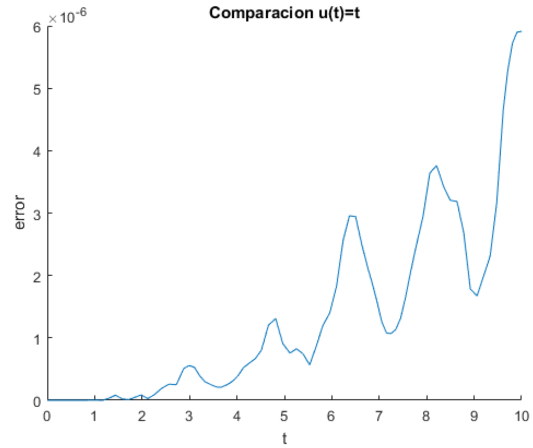

Comparacion u(t)=1
$\times 10^{-6}$
error
t

u(t)=t:

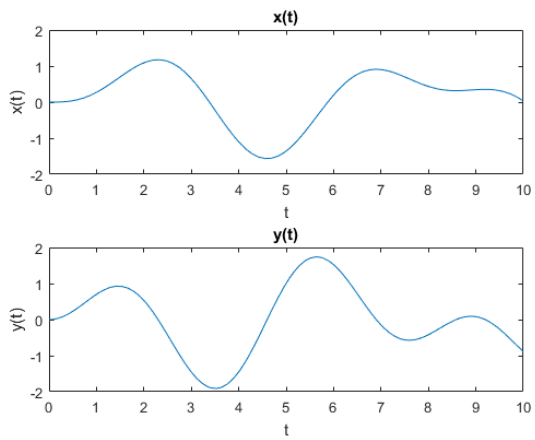

x(t)
x(t)
t
y(t)
y(t)
t

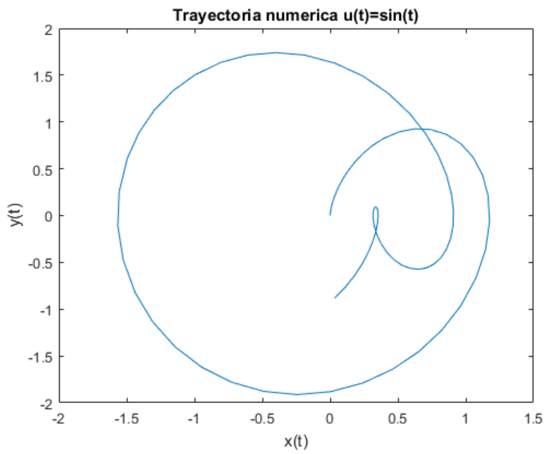

Trayectoria numerica u(t)=t
y(t)
x(t)

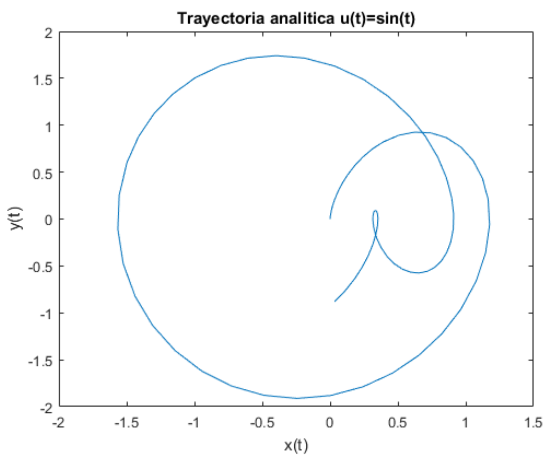

Trayectoria analitica u(t)=t
y(t)
x(t)

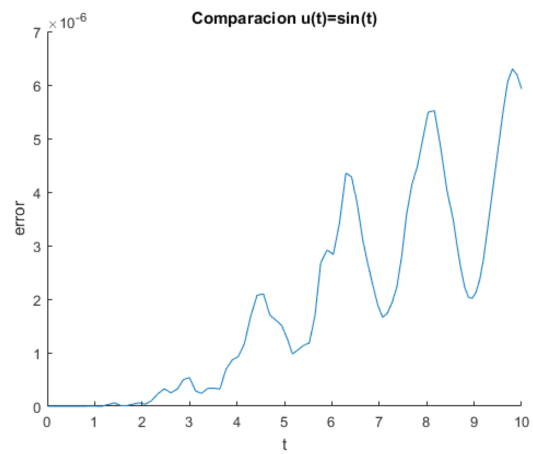

Comparacion u(t)=t
× 10^-6
error
t

u(t)=sin(t):


x(t)
x(t)
t
y(t)
y(t)
t


Trayectoria numerica u(t)=sin(t)
y(t)
x(t)


Trayectoria analitica u(t)=sin(t)
y(t)
x(t)


Comparacion u(t)=sin(t)
× 10^-6
error
t

1.3. Ejercicio 3

Resolvemos el problema de programación lineal con los métodos interior-point y dual-simplex mediante la función linprog()[1]. La siguiente tabla muestra los resultados obtenidos con cada método:

	Interior-Point	Dual-Simplex	Error [$\times 10^{-15}$]
x	0.0000	0.0000	0.0005
y	6.6667	6.6667	0
z	6.0667	6.0667	0.1332
óptimo	-64.8667	-64.8667	0

1.4. Ejercicio 4

Notando que podemos reescribir el problema como uno de minimizacion definimos los parametros y se los entregamos a fmincon():

Código 1: Ejercicio 4

```
% P4
clear all

f = @(x) -x^3; % max Vol = - min -Vol
A = 5;
b = 97;

x = fmincon(f, 1, A, b);

```

Donde el óptimo se alcanza en $x = 19,4$ y donde el valor óptimo es $f(19,4) = 7{,}3014 \times 10^3$.

1.5. Ejercicio 5

Se definen los parametros y la función a utilizar y se entregan a la función fminimax():

Código 2: Ejercicio 4

```
% P5
clear all

lb = [-1,-1];
ub = [0,0];

[x,fval] = fminimax(@myfun,[0.1,0.1], [], [], [], [], lb, ub);

function f = myfun(x)
    f(1) = x(1)^3 + x(2)^3;
    f(2) = x(1) - x(2);
    f(3) = x(1) + x(2) + 7;
```

```
        end

```

Donde el óptimo es $(-1,-1)$.

2. Bocop

2.1. Ejercicio 6

El problema escrito de la forma (M) es como sigue:

$$
(M)\begin{cases}
\min\limits_{u(\cdot)} & m(t_f) & \\
& \begin{pmatrix} \dot{r}(t) \\ v(t) \\ m(t) \end{pmatrix} = \begin{pmatrix} v \\ \frac{1}{r^2} + \frac{1}{2}(T_{max}u - D(r,v)) \\ -bu \end{pmatrix} & t \in [0,t_f] \\
& \begin{pmatrix} 1 \\ 0 \\ 1 \\ 1{,}01 \end{pmatrix} \leq \begin{pmatrix} r(0) \\ v(0) \\ m(0) \\ r(t_f) \end{pmatrix} \leq \begin{pmatrix} 1 \\ 0 \\ 1 \\ 1{,}01 \end{pmatrix} & t \in [0,t_f] \\
& \begin{pmatrix} -\infty \\ -\infty \\ 0 \end{pmatrix} \leq \begin{pmatrix} r \\ v \\ m \end{pmatrix} \leq \begin{pmatrix} +\infty \\ +\infty \\ +\infty \end{pmatrix} & t \in [0,t_f] \\
& 0 \leq u \leq 1 & t \in [0,t_f] \\
& -\infty \leq D(r,v) \leq C & t \in [0,t_f]
\end{cases}
$$

Donde Identificamos en la formulación original como sigue:

$$J(t_0, y(t_0), t_f, y(t_f)) = m(t_f)$$

$$f(y(t), y(t)) = \begin{pmatrix} v \\ \frac{1}{r^2} + \frac{1}{2}(T_{max}u - D(r, v)) \\ -bu \end{pmatrix}$$

$$\Phi(t_0, y(t_0), t_f, y(t_f)) = \begin{pmatrix} r(0) \\ v(0) \\ m(0) \\ r(t_f) \end{pmatrix}$$

$$g(y(t_f), u(t)) = D(r, v)$$

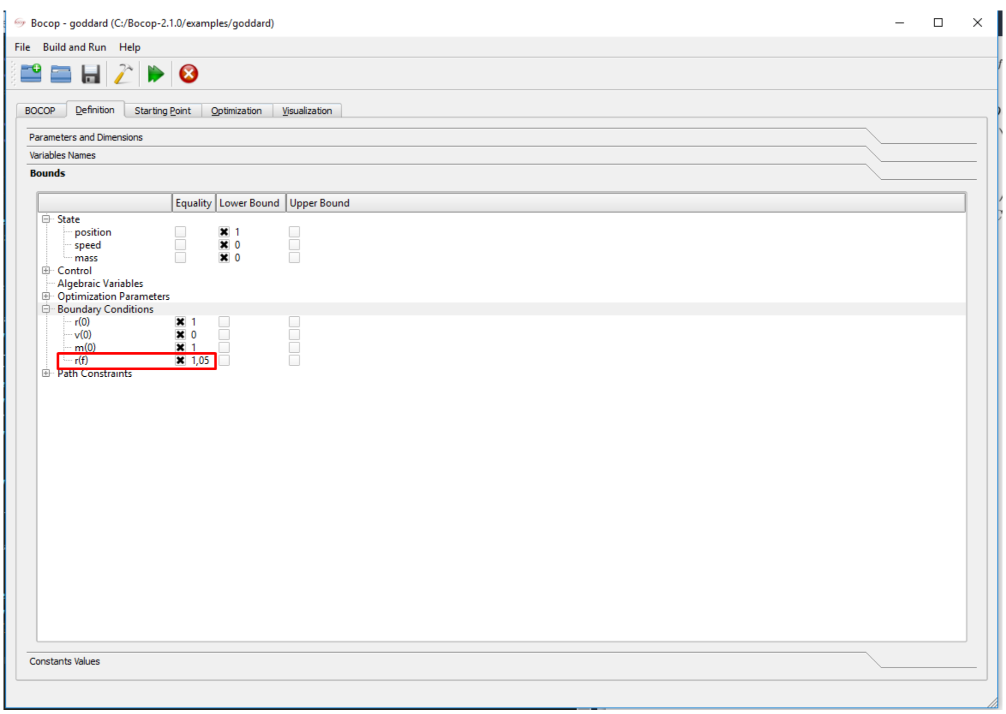
$$\Phi_l = \Phi_u = \begin{pmatrix} 1 \\ 0 \\ 1 \\ 1,01 \end{pmatrix}$$

$$g_l = -\infty; \quad g_u = C$$

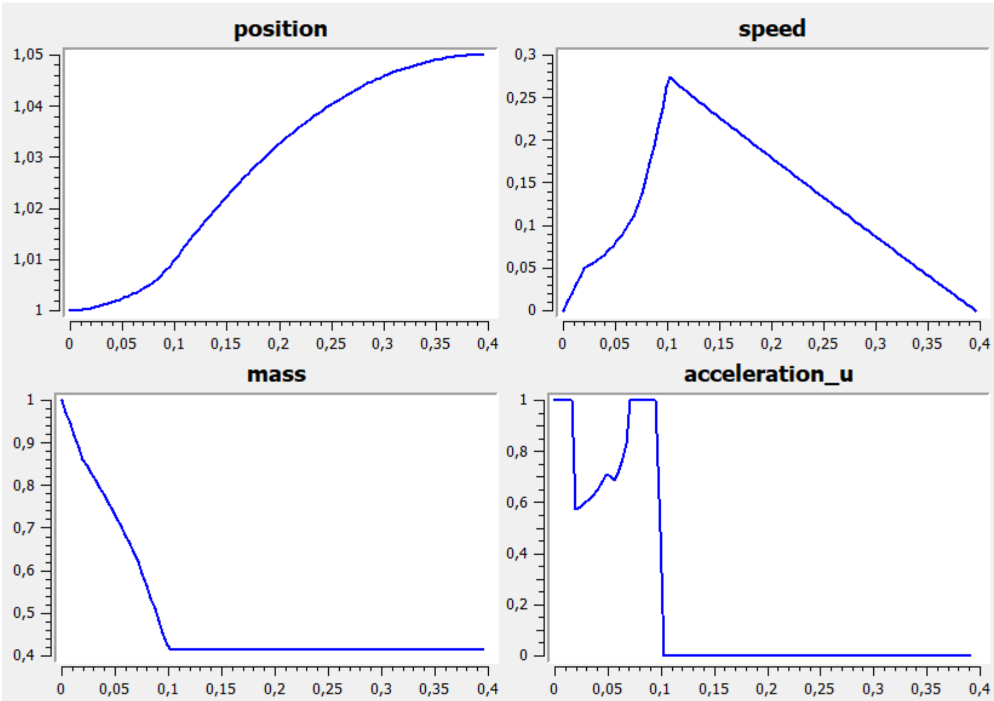
$$y_l = \begin{pmatrix} -\infty \\ -\infty \\ -\infty \end{pmatrix}; \quad y_u = \begin{pmatrix} \infty \\ \infty \\ \infty \end{pmatrix}$$

$$u_l = 0; \quad u_u = 1$$

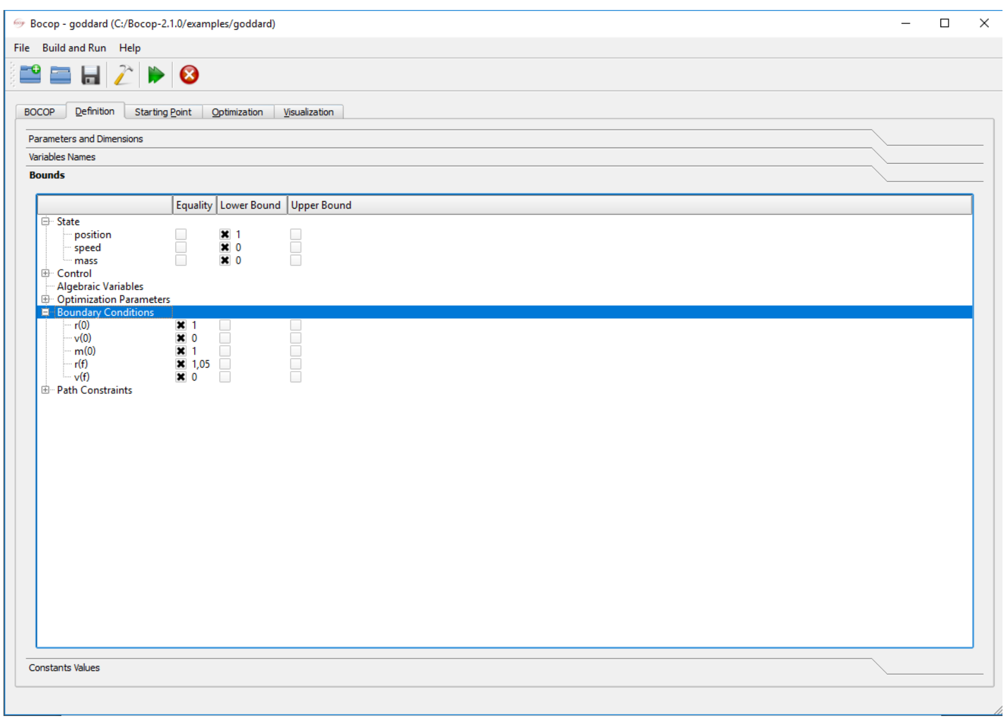
Modificando el problema para considerar $r(t_f) = 1,05$ de la siguiente forma

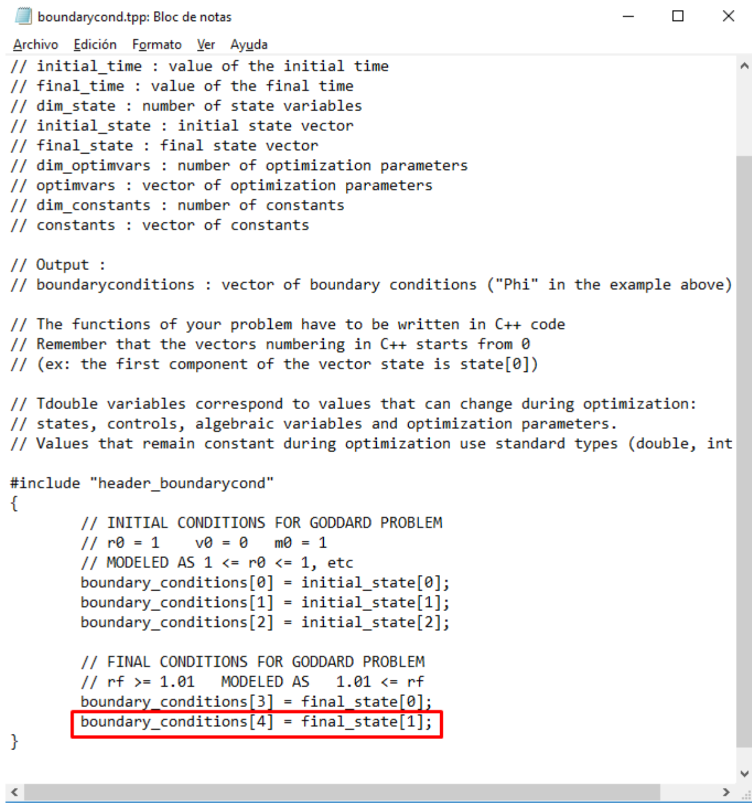


Se obtienen los siguientes resultados



Luego, para considerar $v(t_f) = 0$ realizamos los siguientes pasos:



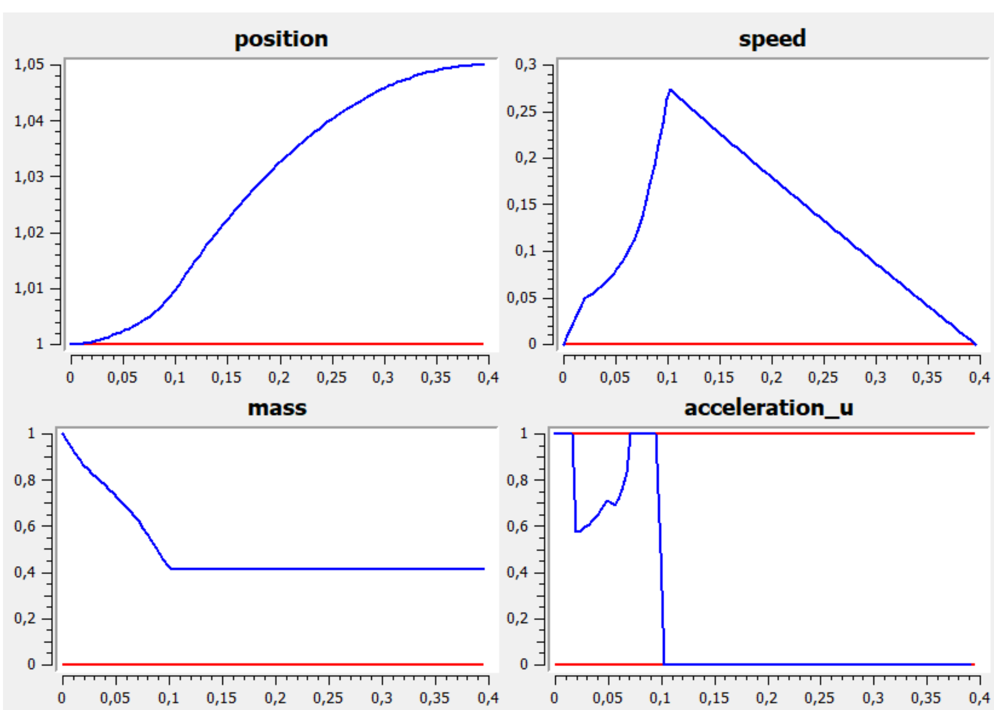


boundarycond.tpp: Bloc de notas

Archivo Edición Formato Ver Ayuda

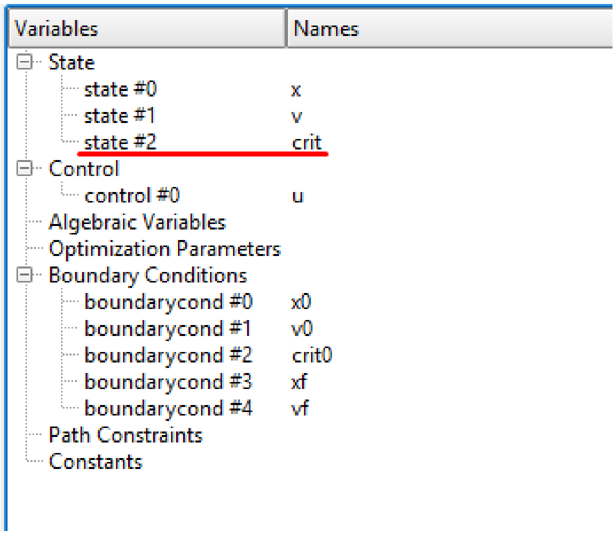
```
// initial_time : value of the initial time
// final_time : value of the final time
// dim_state : number of state variables
// initial_state : initial state vector
// final_state : final state vector
// dim_optimvars : number of optimization parameters
// optimvars : vector of optimization parameters
// dim_constants : number of constants
// constants : vector of constants

// Output :
// boundaryconditions : vector of boundary conditions ("Phi" in the example above)

// The functions of your problem have to be written in C++ code
// Remember that the vectors numbering in C++ starts from 0
// (ex: the first component of the vector state is state[0])

// Tdouble variables correspond to values that can change during optimization:
// states, controls, algebraic variables and optimization parameters.
// Values that remain constant during optimization use standard types (double, int

#include "header_boundarycond"
{
	// INITIAL CONDITIONS FOR GODDARD PROBLEM
	// r0 = 1    v0 = 0   m0 = 1
	// MODELED AS 1 <= r0 <= 1, etc
	boundary_conditions[0] = initial_state[0];
	boundary_conditions[1] = initial_state[1];
	boundary_conditions[2] = initial_state[2];

	// FINAL CONDITIONS FOR GODDARD PROBLEM
	// rf >= 1.01   MODELED AS   1.01 <= rf
	boundary_conditions[3] = final_state[0];
	boundary_conditions[4] = final_state[1];
}
```

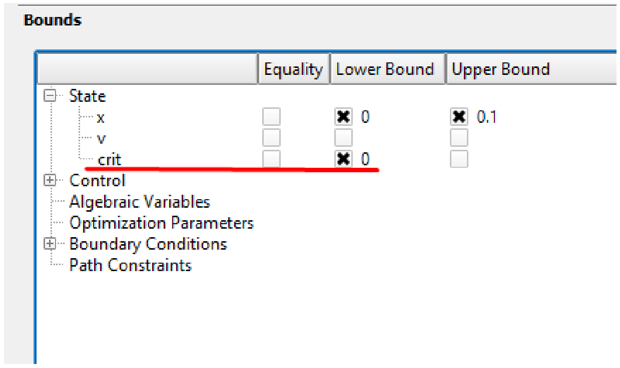
Y los resultados que obtenemos son los mismos, yaque el óptimo ya era llegar con velocidad cero, lo cuál viene de querer minimizar el gasto de combustible.



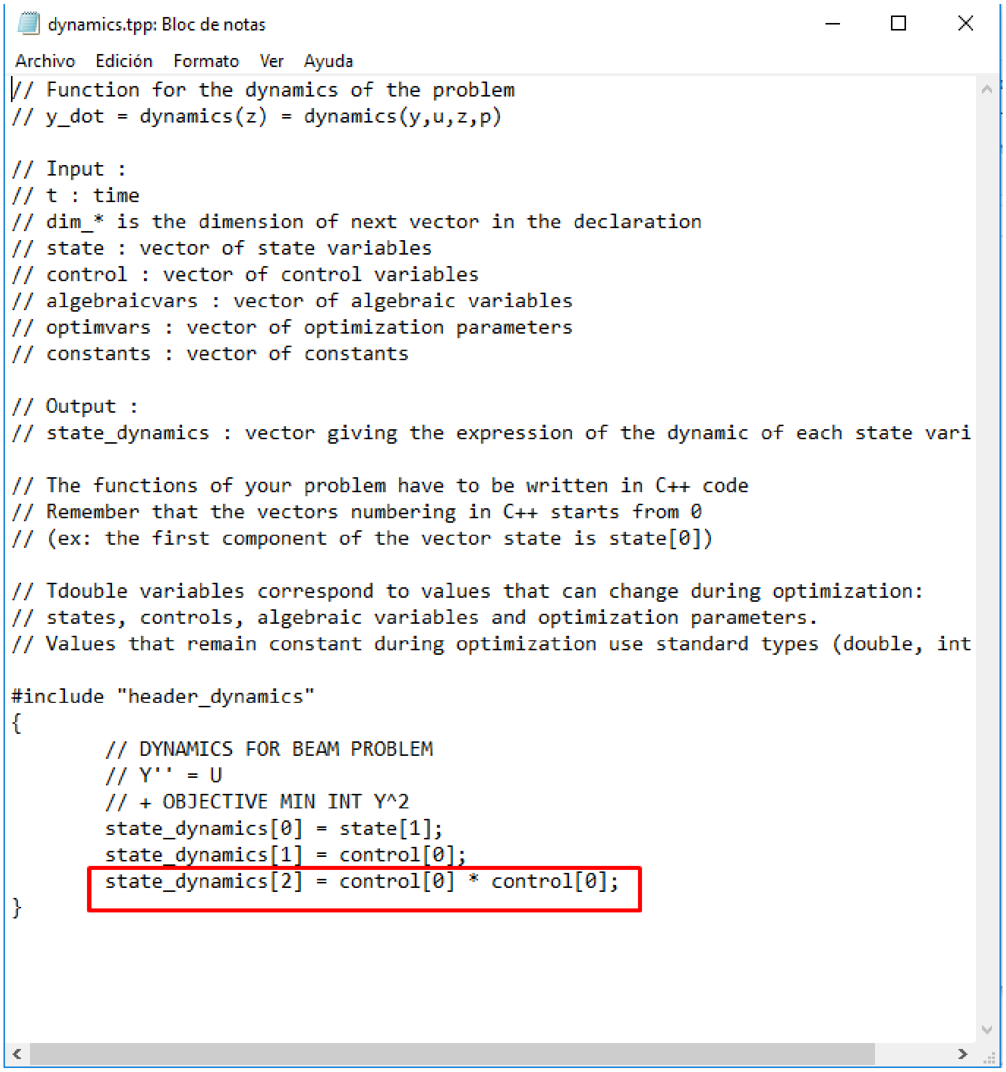
2.2. Ejercicio 7

Notamos que la variable $z(t_f)$ aparece en el ejemplo bajo el nombre de crit, la cuál al estar definida por una integral de números positivos cumple que $z(t_f) \geq 0$,





Luego, notamos que por TFC la dinamica debe ser $z(\dot{t}_f) = u(t_f)^2$, lo cuál se ve en BOCOP como muestra la siguiente imagen:




```
// Function for the dynamics of the problem
// y_dot = dynamics(z) = dynamics(y,u,z,p)

// Input :
// t : time
// dim_* is the dimension of next vector in the declaration
// state : vector of state variables
// control : vector of control variables
// algebraicvars : vector of algebraic variables
// optimvars : vector of optimization parameters
// constants : vector of constants

// Output :
// state_dynamics : vector giving the expression of the dynamic of each state vari

// The functions of your problem have to be written in C++ code
// Remember that the vectors numbering in C++ starts from 0
// (ex: the first component of the vector state is state[0])

// Tdouble variables correspond to values that can change during optimization:
// states, controls, algebraic variables and optimization parameters.
// Values that remain constant during optimization use standard types (double, int

#include "header_dynamics"
{
	// DYNAMICS FOR BEAM PROBLEM
	// Y'' = U
	// + OBJECTIVE MIN INT Y^2
	state_dynamics[0] = state[1];
	state_dynamics[1] = control[0];
	state_dynamics[2] = control[0] * control[0];
}
```

El problema escrito de la forma (M) es como sigue:

$$
(M)\begin{cases}
\min\limits_{u(\cdot)} & z(t_f) \\
\dot{\begin{pmatrix} x(t) \\ v(t) \\ z(t) \end{pmatrix}} = \begin{pmatrix} v \\ u(t) \\ \frac{1}{2}u(t)^2 \end{pmatrix} & t \in [0, t_f] \\
\begin{pmatrix} 0 \\ 0 \\ 1 \\ -1 \end{pmatrix} \leq \begin{pmatrix} x(0) \\ x(t_f) \\ v(0) \\ v(t_f) \end{pmatrix} \leq \begin{pmatrix} 0 \\ 0 \\ 1 \\ -1 \end{pmatrix} & \\
\begin{pmatrix} -\infty \\ -\infty \\ 0 \end{pmatrix} \leq \begin{pmatrix} x(t) \\ v(t) \\ z(t) \end{pmatrix} \leq \begin{pmatrix} a \\ +\infty \\ +\infty \end{pmatrix} & t \in [0, t_f] \\
-10 \leq u(t) \leq 10 & t \in [0, t_f]
\end{cases}
$$

Donde Identificamos en la formulación original como sigue:

$$
\begin{aligned}
&J(t_0, y(t_0), t_f, y(t_f)) = z(t_f) \\
&f(y(t), y(t)) = \begin{pmatrix} v \\ u(t) \\ \frac{1}{2}y(t)^2 \end{pmatrix} \\
&\Phi(t_0, y(t_0), t_f, y(t_f)) = \begin{pmatrix} x(0) \\ x(t_f) \\ v(0) \\ v(t_f) \end{pmatrix} \\
&g(y(t_f), u(t)) = 0 \\
&\Phi_l = \Phi_u = \begin{pmatrix} 0 \\ 0 \\ 1 \\ -1 \end{pmatrix} \\
&g_l = 0; \;\; g_u = 0 \\
&y_l = \begin{pmatrix} -\infty \\ -\infty \\ 0 \end{pmatrix}; \;\; y_u = \begin{pmatrix} a \\ +\infty \\ +\infty \end{pmatrix} \\
&u_l = -10; \;\; u_u = -10
\end{aligned}
$$

Resolvemos este problema en BOCOP con parametro $a = 0,05$ y $a = 0,2$, usando la variable $\mathrm{crit}(t) = 2z(t)$:

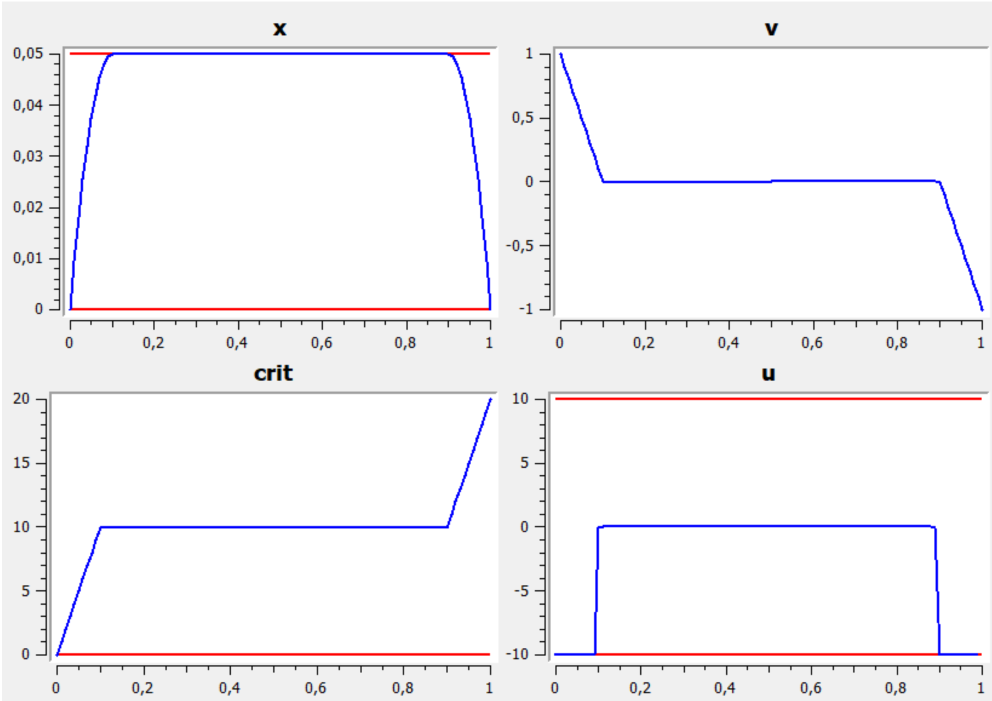


Figura 1: $a = 0,05$

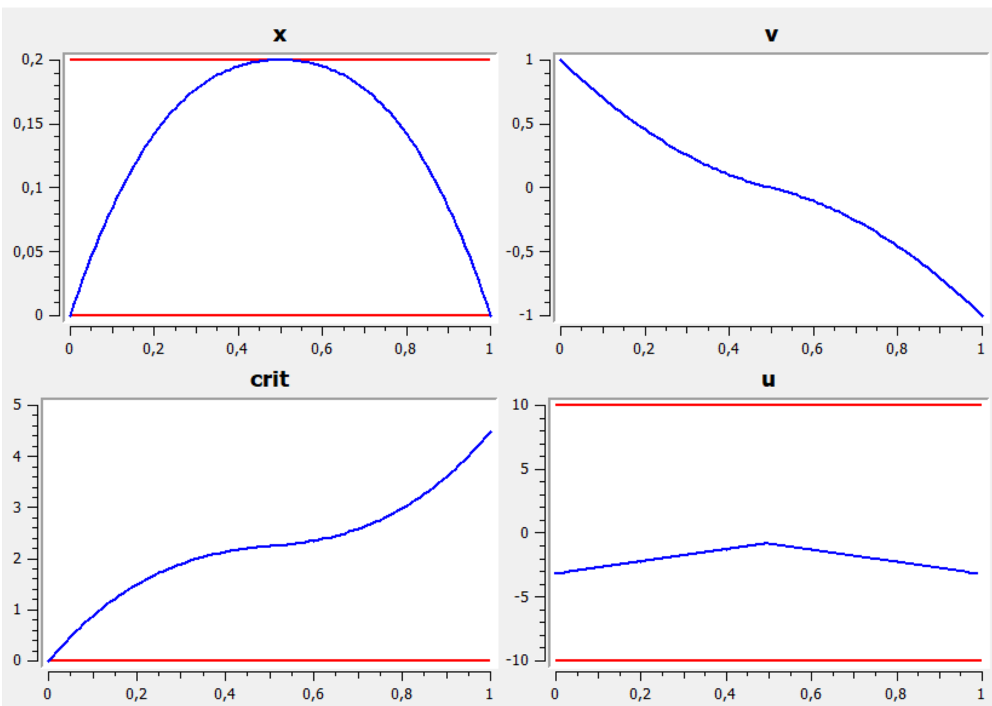


Figura 2: $a = 0,2$

Referencias

[1] Documentación de software Matlab
https://la.mathworks.com/help/index.html