Universidad de Chile
Facultad de Ciencias Físicas y Matemáticas
Departamento de Ingeniería Matemática
MA4703-1 Control Óptimo: Teoría y Laboratorio

Laboratorio 3

Problemas de Tiempo Mínimo

Integrantes: Diego Dominguez
Arturo Lazcano
Profesor: Héctor Ramírez
Auxiliares: Javier Madariaga
Pablo Araya

Fecha de entrega: 4 de Octubre de 2022

Índice de Contenidos

1. Ejercicio 1

Del sistema planteado con las matrices

El sistema original del circuito es:

$$\begin{aligned} L_1\frac{di_i}{dt}(t) + K\frac{di_2}{dt}(t) + R_c i_1(t) &= u(t) \\ K\frac{di_i}{dt}(t) + L_2\frac{di_2}{dt}(t) + R_w i_2(t) &= 0 \\ i_1(0) &= -i_0 \\ i_2(0) &= 0 \\ u(t) &\in [-a, a]\ \forall t \geq 0 \end{aligned}$$

Las primeras dos ecuaciones de este sistema equivalen a:

$$\begin{aligned} L_1 i_1' + K i_2' + R_c i_1 &= u \\ K i_1' + L_2 x_2' R_w i_2 &= 0 \end{aligned}$$

Luego,

$$\begin{aligned} i_1' &= \frac{u - K i_2' - R_c i_1}{L_1} = \frac{-L_2 i_2' - R_w i_2}{L_2} \\ i_2' &= \frac{u - L_1 i_1' - R_c i_1}{K} = \frac{-K i_1' - R_w i_2}{L_2} \end{aligned}$$

Por lo que reemplazando i_2 en la primera ecuación e i_1 en la segunda resulta:

$$i_1' = \frac{L_2 u + K^2 i_1' + K R_w i_2 - R_c L_2 i_1}{L_1 L_2}; \quad i_2' = \frac{uK + L_1 L_2 i_2' + L_1 R_w i_2 - R_c K i_1}{K^2}$$

Así,

$$\begin{aligned} i_1'(L_1 L_2 - K^2) &= L_2 u + K R_w i_2 - R_c L_2 i_1 \\ i_2'(K^2 - L_1 L_2) &= uK + L_1 R_w i_2 - R_c K i_1 \end{aligned}$$

Por último, multiplicando la segunda ecuación por -1 nos queda el siguiente sistema matricial:

$$\begin{pmatrix} i_1' \\ i_2' \end{pmatrix} = \frac{1}{(1-\alpha^2)L_1 L_2} \begin{pmatrix} -L_2 R_C & \alpha\sqrt{L_1 L_2} R_w \\ \alpha\sqrt{L_1 L_2} R_c & -L_1 R_w \end{pmatrix} \begin{pmatrix} i_1 \\ i_2 \end{pmatrix} + \frac{1}{(1-\alpha^2)L_1 L_2} \begin{pmatrix} L_2 \\ -K \end{pmatrix} u(t)$$

2. Ejercicio 2

Para mostrar numéricamente que el sistema es controlable, se calcula la matriz de Kalman con el uso del toolbox **control**. La matriz obtenida es

$$Kalman = \begin{pmatrix} -0.42826552 & 3.38265364 \\ 0.50988783 & -4.79217888 \end{pmatrix}$$

Así, la matriz de Kalman obtenida es de rango completo, pues las dos columnas obtenidas son linealmente independientes entre sí por lo cual es sistema es controlable.

3. Ejercicio 3

Para ver que las variables i_1' e i_2' tienden a 0 cuando el tiempo tiende a infinito y $u(t)$ es constante, se grafican varios ejemplos de controles.

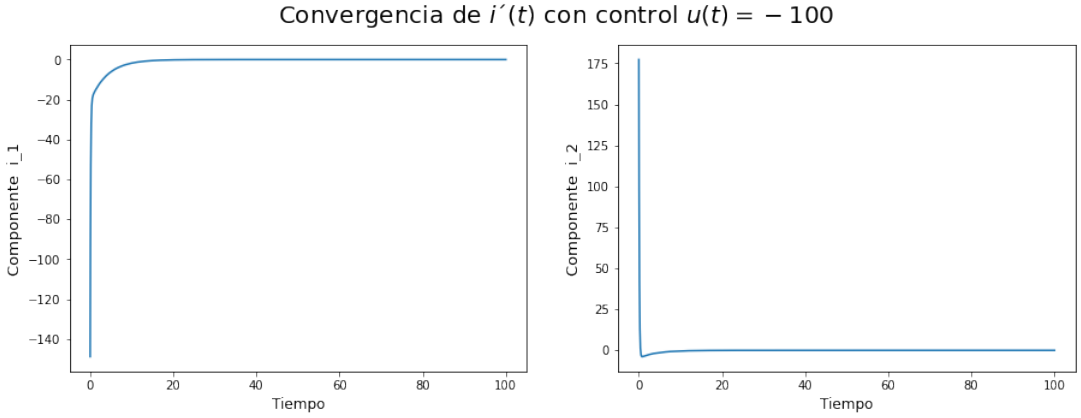


Figura 1: Derivada de corrientes eléctricas con respecto al tiempo con u=-100

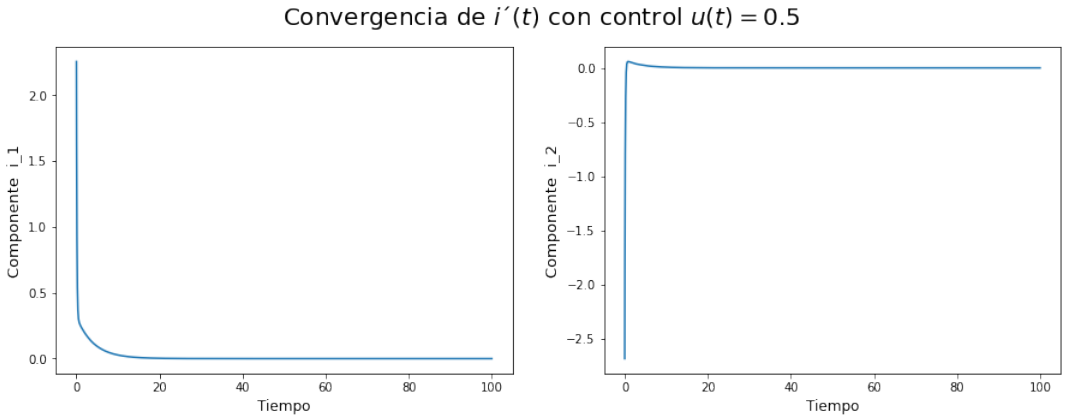


Figura 2: Derivada de corrientes eléctricas con respecto al tiempo con u=0.5

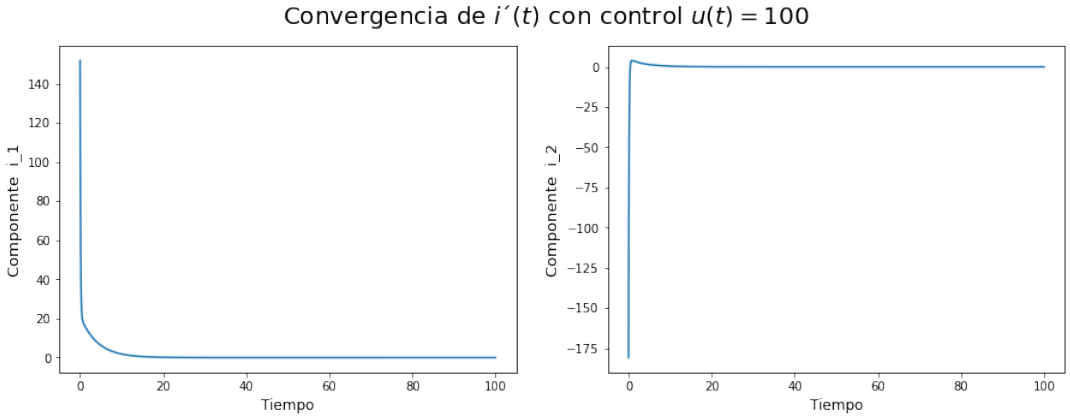


Figura 3: Derivada de corrientes eléctricas con respecto al tiempo con u=100

4. Ejercicio 4

El sistema original escrito como problema de control óptimo de tiempo mínimo es:

$$\begin{aligned}
\min_{u} \quad & T \\
\text{s.a.} \quad & i_1' = \frac{1}{(1-\alpha^2)L_1L_2}(-L_2R_ci_1 + \alpha\sqrt{L_1L_2}R_wi_2 + L_2u) \\
& i_2' = \frac{1}{(1-\alpha^2)L_1L_2}(\alpha\sqrt{L_1L_2}R_ci_1 - L_1R_wi_2 - Ku) \\
& (i_1(0), i_2(0)) = (-i_0, 0) \\
& (i_1(T), i_2(T)) = (i_0, 0) \\
& |u(\cdot)| \leq a
\end{aligned}$$

5. Ejercicio 5

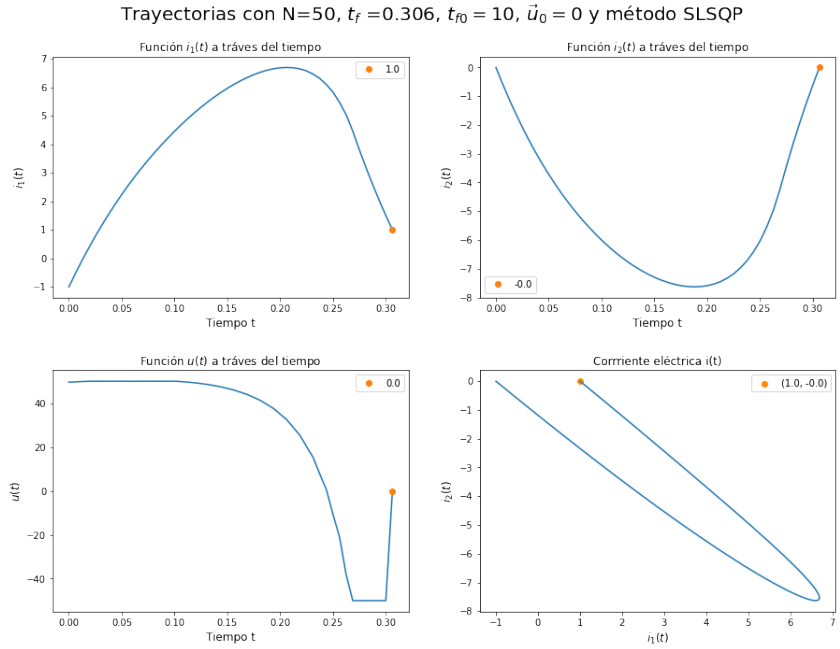
Dado un $t_f > 0$ fijo y discretizando la dinámica del problema en N puntos mediante el método de Euler, se obtiene la siguiente iteración para la corriente eléctrica:

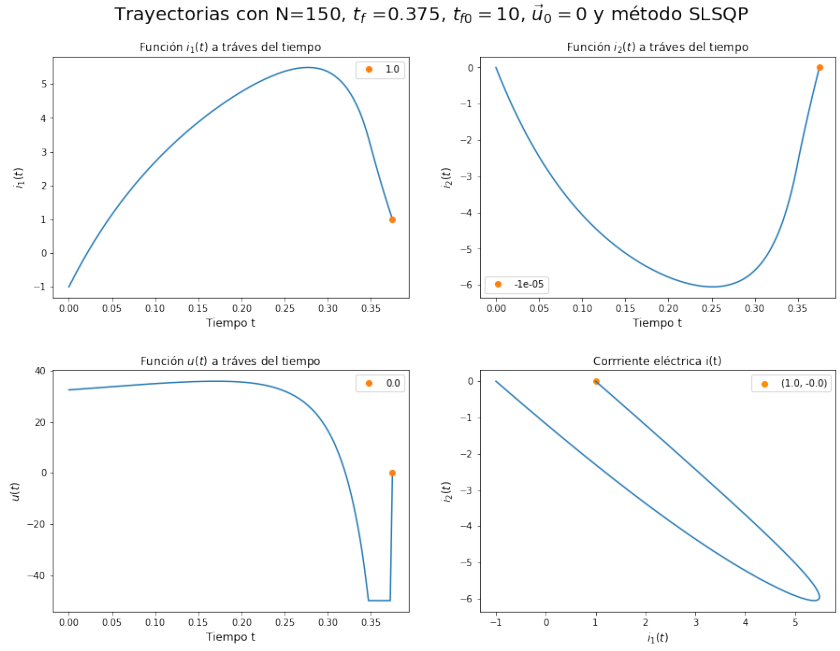
$$\begin{aligned}
i_1^{j+1} &= i_1^j + \Delta_{t_f}\frac{1}{(1-\alpha^2)L_1L_2}(-L2_{Rc}i_1^j + \alpha\sqrt{L_1L_2}R_wi_2^j + L_2u(j)) \\
i_2^{j+1} &= i_2^j + \Delta_{t_f}\frac{1}{(1-\alpha^2)L_1L_2}(\alpha\sqrt{(L_1L_2)}R_ci_1^j - L_1R_wi_2^j - Ku(j))
\end{aligned}$$

Para $j \in \{1, ..., N\}$, $(u(j))_{j=1}^N$ y considerando $\Delta_{t_f} = \frac{t_f}{N}$

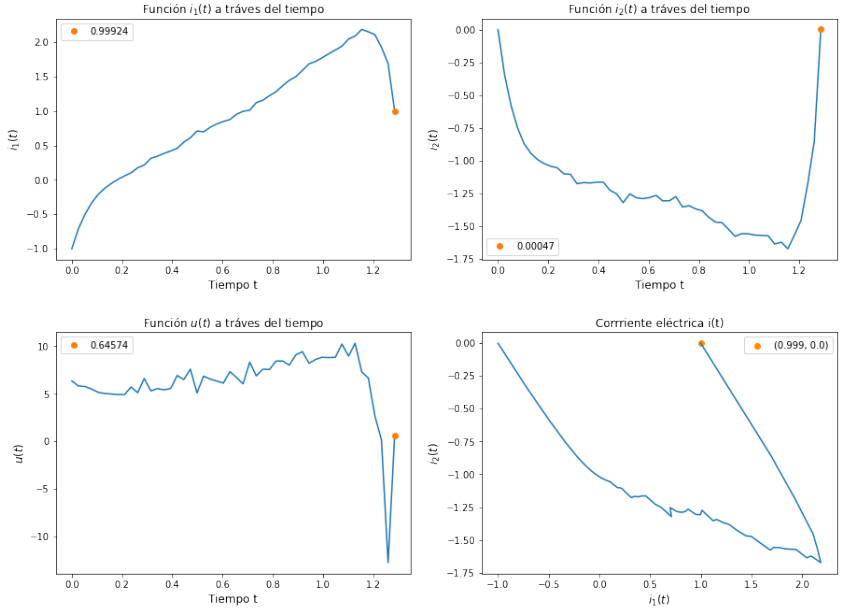
6. Ejercicio 6

Mediante la programación de un problema de optimización no lineal, se puede encontrar el control óptimo a tiempo mínimo usando la discretización anterior al momento de calcular las componentes de la corriente eléctrica $i(t)$. Este proceso se realizó para distintos valores de N y variando los métodos de optimización de la función **minimize** de **scipy**. En particular se utilizó $N = 50, 100, 150$ y los métodos SLSQP, COBYLA, **trust-constr**.

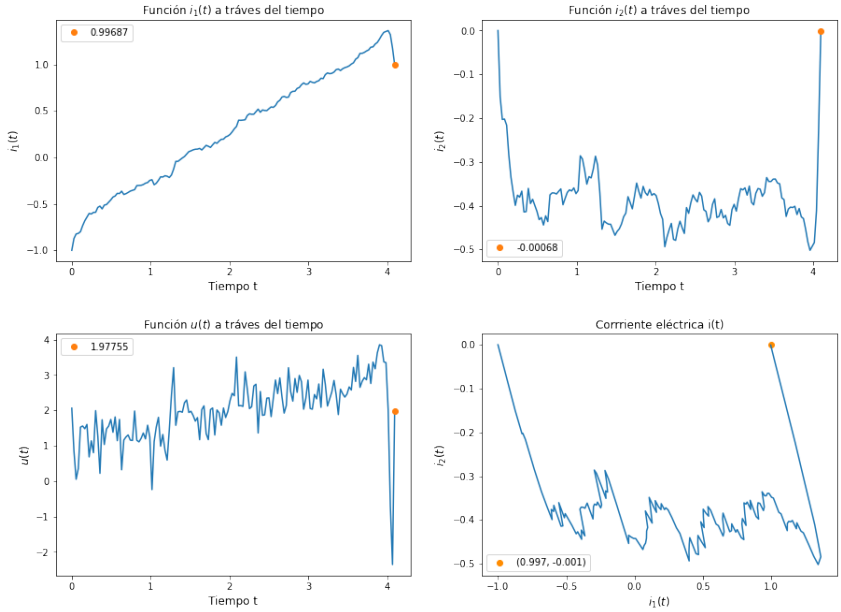


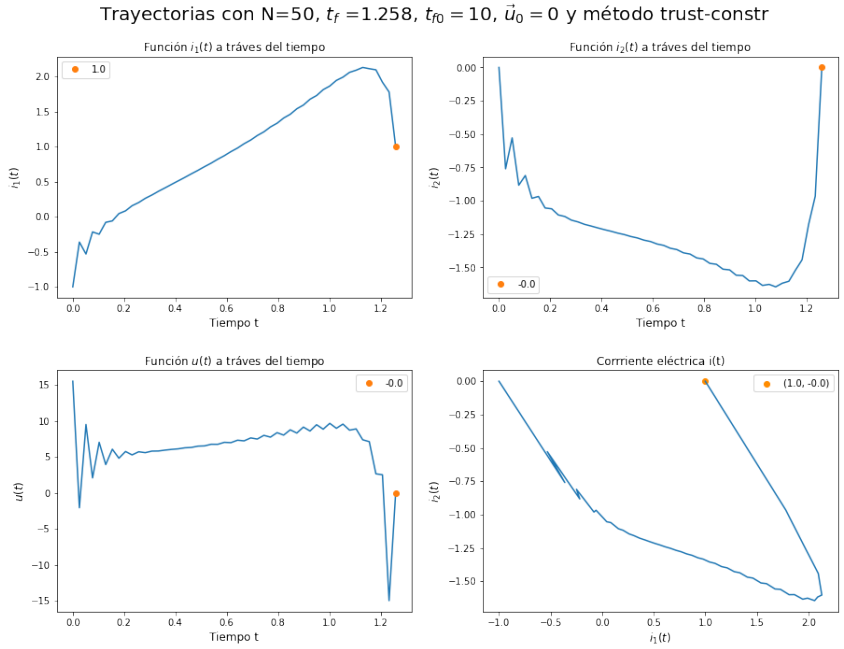


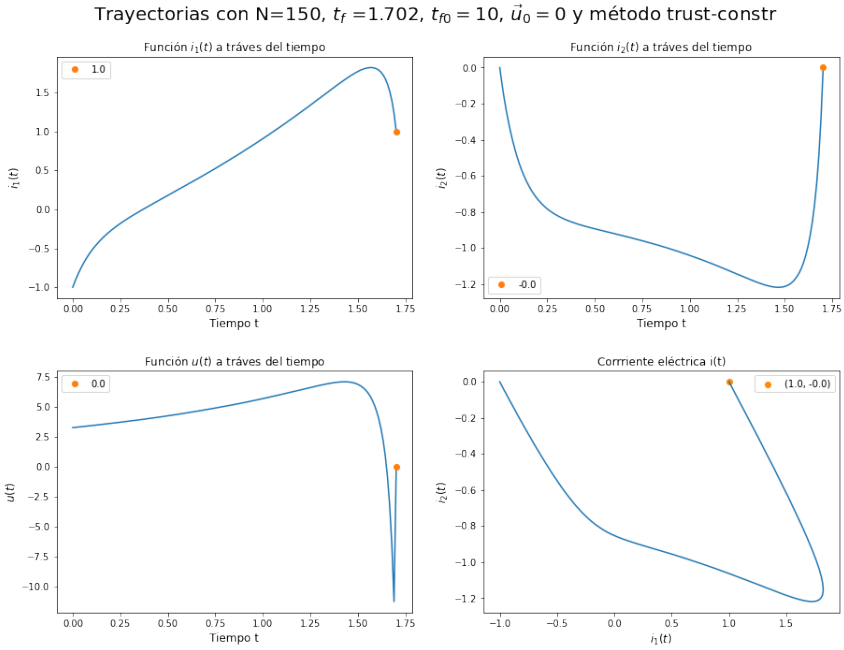
Trayectorias con N=50, t_f =1.285, $t_{f0} = 10$, $\vec{u}_0 = 0$ y método COBYLA



Trayectorias con N=150, t_f =4.095, $t_{f0} = 10$, $\vec{u}_0 = 0$ y método COBYLA







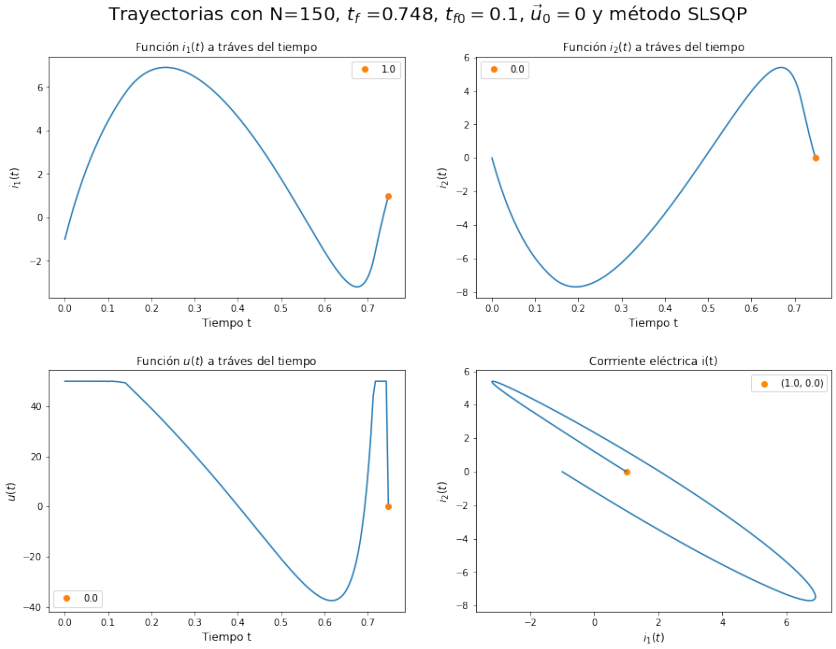
Se observa que para el método SLSQP basta 50 iteraciones para obtener trayectorias suaves tanto para la corriente como para el control, por lo que con $N = 150$ se obtiene resultados muy similares. Por otro lado, con el método COBYLA se observa que valor objetivo empeora a medida que aumenta N. Por último, el método trust-constr entrega soluciones con muy poco ruido con $N = 50$, y el resultado mejora cuando N aumenta. Aún así, el tiempo final obtenido es mayor al que entrega SLSQP, por lo tanto, para un $t_0 = 0$ y $(u_0(j))_{j=1}^{N} = \vec{0}$, se concluye que el tiempo mínimo es 0.3.

Tabla 1: Tiempo de ejecución en segundos para distintos métodos y valores de N.

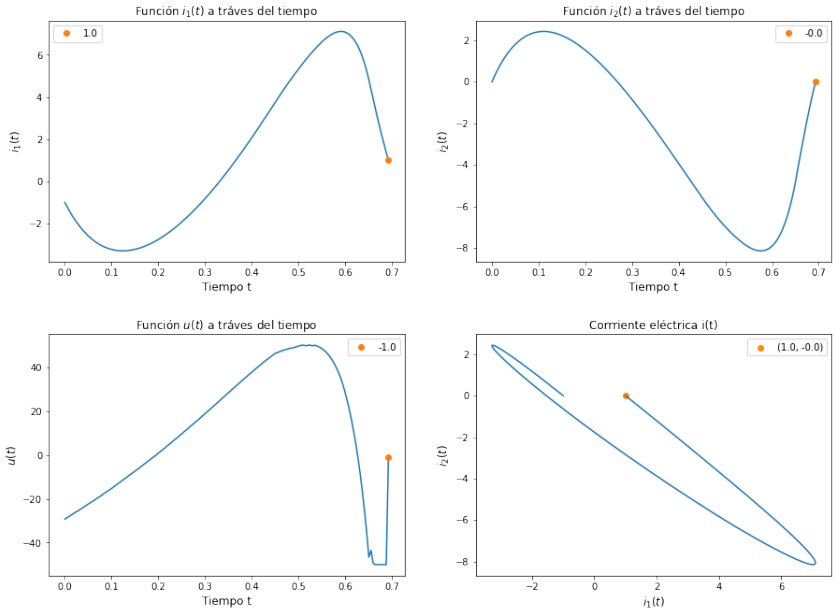
N	SLSQP	COBYLA	trust-constr
50	2.308	1.356388	27.424268
100	6.655	2.630	79.862623
150	14.111	5.696879	161.064205

Notemos que el tiempo de ejecución del problema de minimización crece a medida que aumenta la cantidad de puntos discretizados N. De hecho se observa que por cada 50 discretizaciones extras, el tiempo de ejecución tienede a duplicarse.

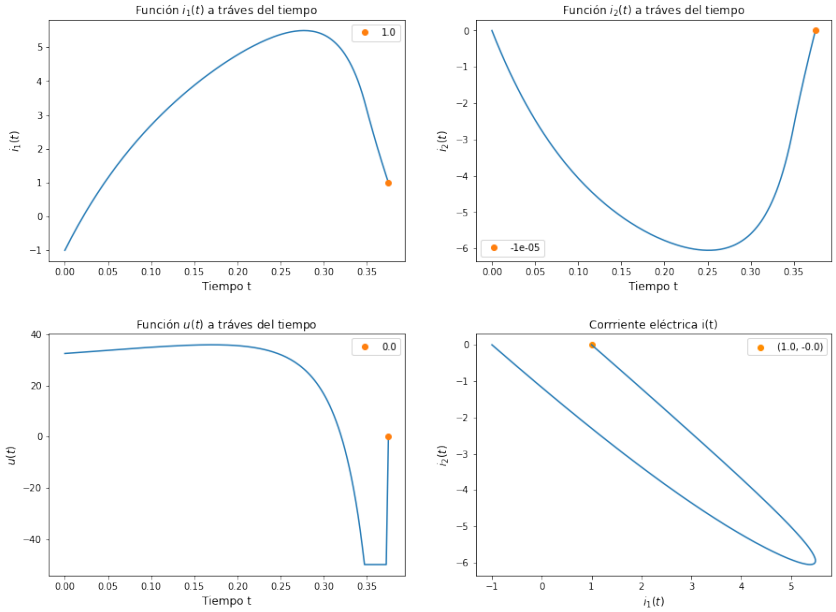
Así que para los experimentos siguientes se fija $N = 150$ y se utilizará el método `SLSQP`, pues con estos paramétros se obtuvo la mejor solución en el menor tiempo de ejecución.



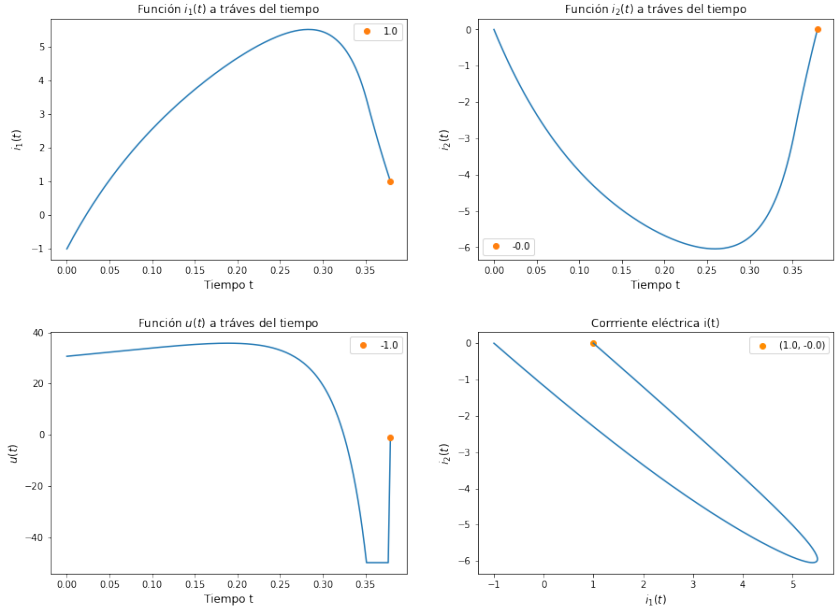
Trayectorias con N=150, t_f =0.692, $t_{f0} = 0.1$, $\vec{u}_0$ = -1 y método SLSQP



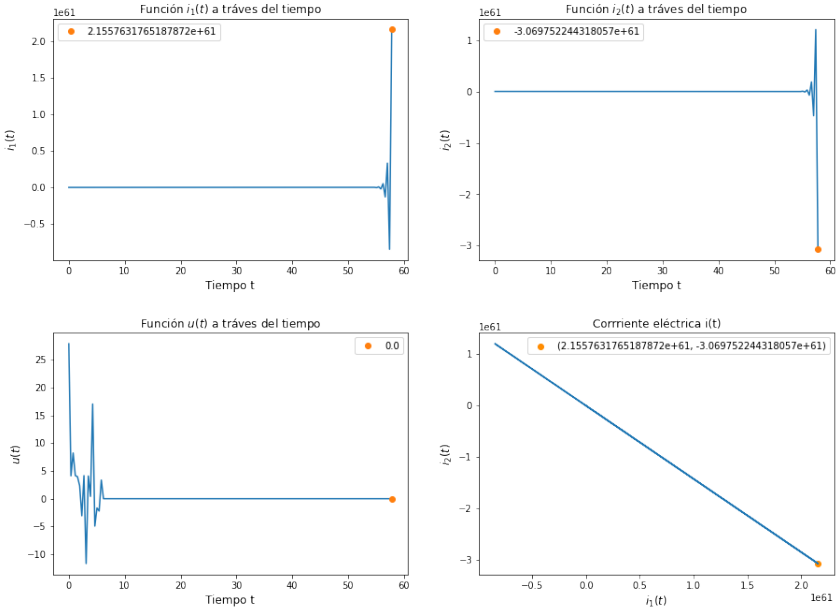
Trayectorias con N=150, t_f =0.375, $t_{f0} = 10$, $\vec{u}_0 = 0$ y método SLSQP

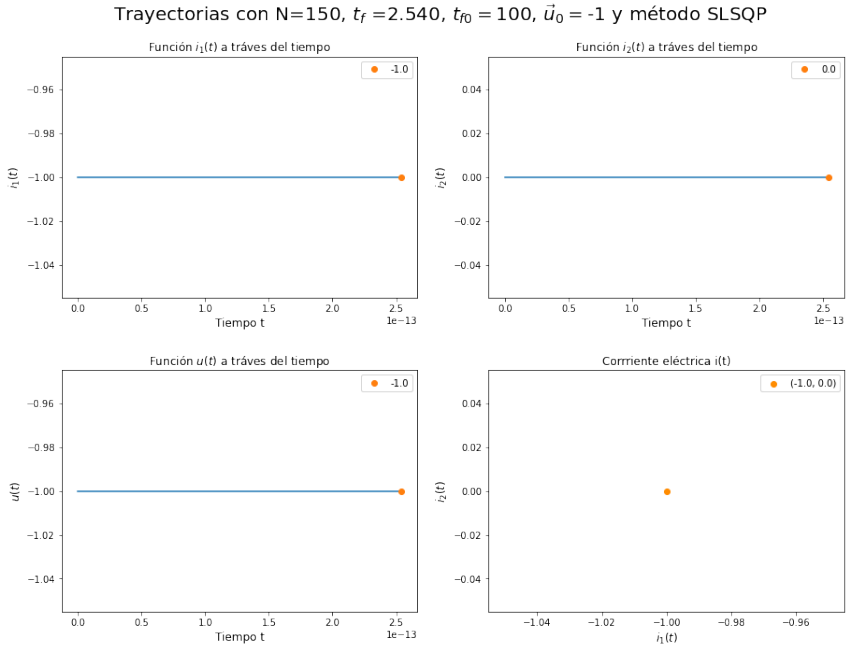


Trayectorias con N=150, t_f =0.379, $t_{f0} = 10$, $\vec{u}_0 = -1$ y método SLSQP



Trayectorias con N=150, t_f =57.80, $t_{f0} = 100$, $\vec{u}_0 = 0$ y método SLSQP





Para los parámetros escogidos, se observa que para tiempos t_{f_0} pequeños, el control óptimo obtenido es bastante similar y conserva un una evolución bastante sueva durante todo su trayecto. Solo se observa una pequeña diferencia en el trayecto de las corrientes al variar el vector de control inicial $\vec{u}_0$. Por otro lado, para tiempos muy grandes (como 100), el problema converge a un punto que no es factible, dando como resultado trayectos erráticos tanto para el control como para la corriente elétrica.

7. Ejercicio 7

Con el objetivo de resolver el problema con la caracterización de control extremal, se escribe un sistema extendido que incluya a i y a p en su variable, definiendo nuevas matrices por bloques y usando que $u^*(t) = a \cdot sgn(p(t)^T B))$.
Luego, se crea una función F que reciba como parámetros tf y h y entregue las posiciones finales de i_1 e i_2.

Por último, se buscan los ceros de la función F con ayuda de scipy y así, encontrar el tiempo final tf y el valor h asociado a la condición inicial $p(0)$ con tal de resolver el problema de minimización en el tiempo.

Por problemas con las funciones de scipy, no se pudo encontrar ceros de la función F, por lo que se deja en el archivo .ipynb el código de esta función y se omiten las comparaciones de tiempo y eficacia del método de resolución indirecto.

8. Ejercicio 8

Las condiciones de borde son 4: 2 de tiempo inicial para i y 2 de tiempo final para p. Por último, el criterio a optimizar es final_time, es decir, problema de tiempo mínimo. Usando el programa BOCOP para resolver el sistema original, se obtienen las siguientes variables de estado y control:

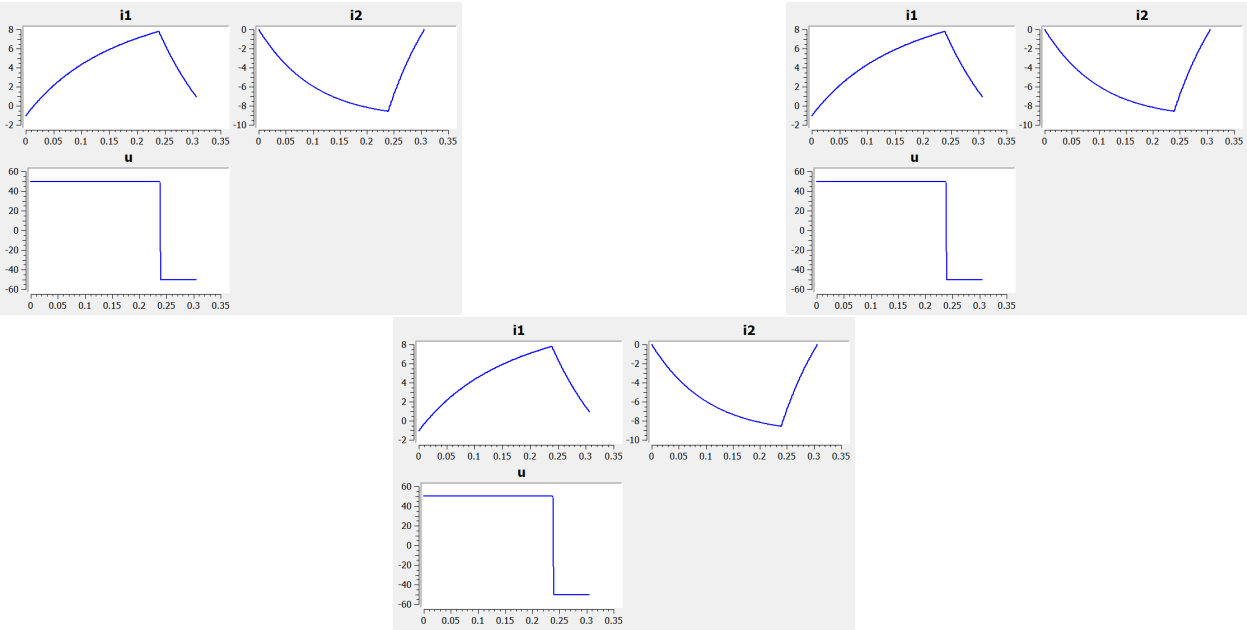


Figura 4: Solución óptima del programa BOCOP usando diferentes métodos

Los métodos usados fueron Midpoint, Gauss y Lobatto mientras que sus tiempos de ejecución fueron 5.21, 12.2 y 47.37 segundos respectivamente y sus valores objetivos fueron todos 0.3 con diferenciados de pocas milésimas.

9. Ejercicio 9

Para finalizar, podemos concluir que tanto la función programada en la pregunta 5 y 6 usando el método SLSQP, como el sistema programado en BOCOP nos entrega un tiempo mínimo de 0.3 con trayectorias similares tanto para el control óptimo como para la corriente eléctrica.
Con respecto al tiempo de ejecución, el método Midpoint de BOCOP es más rápido que SLSQP con N grande.
Por lo tanto, en los problemas de control óptimo, BOCOP nos entrega una buena aproximación de los resultados con una rápida ejecución, de esta forma, nos ahorramos tener que plantear el sistema matricial en python y de escribir el problema de optimización, el cuál es muy sensible a los parámetros de entrada para N pequeños y tarda demasiado cuando se quiere aumentar N para afinar la discretización.