1 Modelado dinámico de la máquina sincrona en el marco de referencia dq0

Uriel Sandoval

Este trabajo reporta los resultados de una simulación dinámica de un máquina síncrona. Los parametros utilizados son los de un hidro generador reportado en libro de Krauze.

Se cargan los datos y se crea la instancia:

```
from sistema_sincrona import Maquina_Sincrona
from datos import cargar_datos
import numpy as np
```

```
dat = cargar_datos('krauze_hidro')
maq = Maquina_Sincrona(dat)
```

Se arranca una simulación con $CI = 0$ pero ya girando a velocidad síncrona para obtener el estado estable:

```
enlaces = [0]*6
velocidad = [2*np.pi*60]
angulo = [0]
maq.x0 = np.array(enlaces+velocidad+angulo)
maq.dinamico(10)
```

Se guardan los últimos estados para utilizarlos como las CI del disturbio:

```
ci = maq.X[-1][:]
```

Se crea el evento que se quiere aplicar. En este caso la falla consiste en aumentar de 0 a par nominal el par mecánico en la flecha del generador:

```
falla = {'tipo': 'mod_par',
        'ti': 1,
        'tf': 1,
        'valor': 1}
```

Se simula nuevamente ya con la falla aplicada al primer segundo:

```
maq.x0 = ci
maq.eventos.append(falla)
maq.dinamico(5)
```

```
(True, 'tf': 1, 'valor': 1, 'tipo': 'mod_par', 'ti': 1)
Se apica evento
(False, 'tf': 1, 'valor': 1, 'tipo': 'mod_par', 'ti': 1)
Se libera evento
```

1.1 Resultados

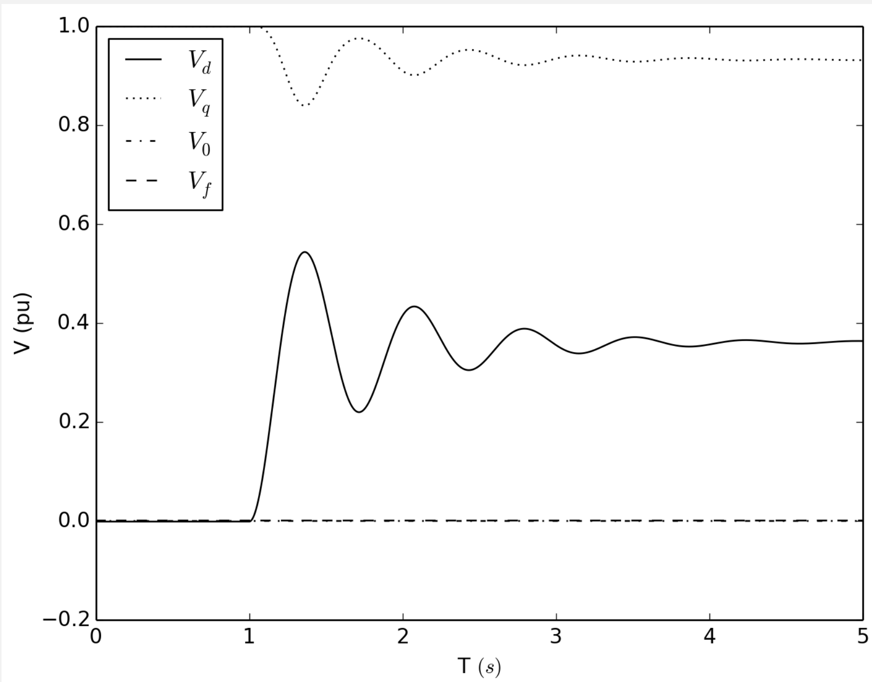
```
%pylab inline
```

```
Populating the interactive namespace from numpy and matplotlib
```

1.1.1 Voltajes

```
estilos = ['k-', 'k:', 'k-.', 'k--']
for p, nombre in enumerate(maq.nombres['V']):
    plot(maq.t, maq.data['V'][:, p], estilos[p],linewidth=1, label='$'+nombre+'$')
legend(loc=0)
xlabel('T $(s)$')
ylabel('V (pu)')
```

```
<matplotlib.text.Text at 0x10a7e2190>
```



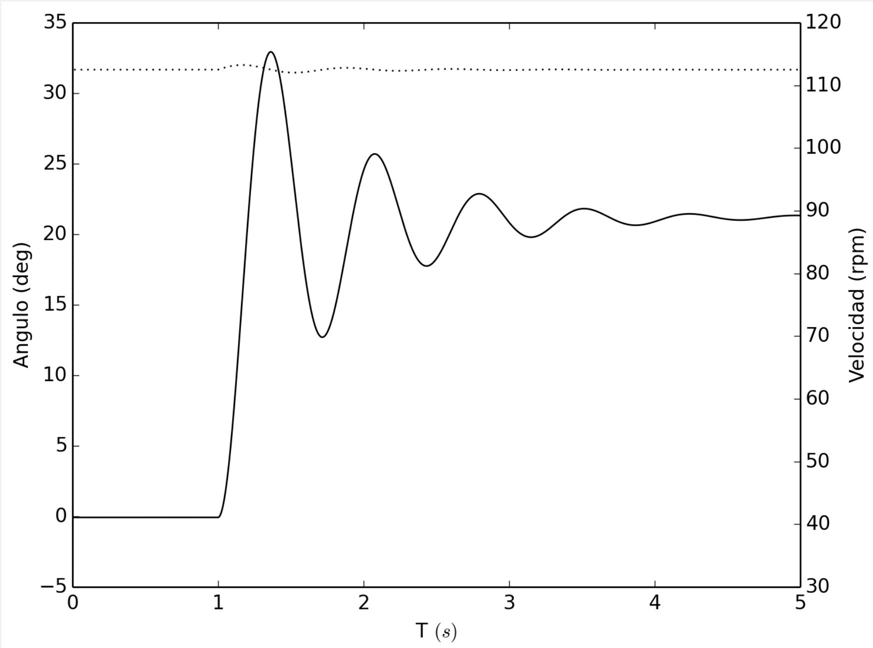
Esta gráfica de los voltajes muestra como hay un caida en el voltaje de eje directa que está relacionado con el voltaje en terminales ABC de la máquina. Eso se podría compensar al agregar un AVR que regulara la excitación.

1.1.2 Velocidad ángulo

```
_, ax1 = subplots()
ax2 = ax1.twinx()
ax1.plot(maq.t, maq.data['d'], 'k-', linewidth=1, label='$\delta$')
```

```
ax2.plot(maq.t, maq.data['w'], 'k:',linewidth=1, label='$\omega$')
ax1.set_xlabel('T $(s)$')
ax1.set_ylabel('Angulo (deg)')
ax2.set_ylabel('Velocidad (rpm)')
```

```
<matplotlib.text.Text at 0x10a8b0990>
```

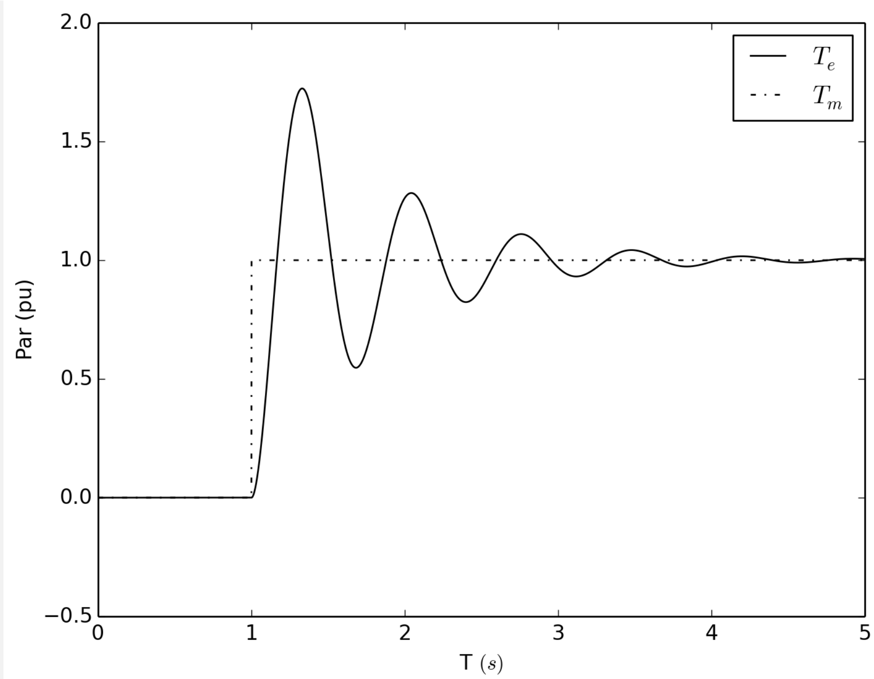


Se nota como el ángulo de la máquina se abre más al estar más carga mecánicamente. Llega hasta un ángulo de 20, el cual está dentro del nominal. La velocidad casi no presenta cambio.

1.1.3 Par eléctrico y mecánico

```
plot(maq.t, maq.data['Te'], 'k-', linewidth=1, label='$T_e$')
plot(maq.t, maq.data['Tm'], 'k-.', linewidth=1, label='$T_m$')
xlabel('T $(s)$')
ylabel('Par (pu)')
legend(loc=0)
```

```
<matplotlib.legend.Legend at 0x10a9d5b50>
```

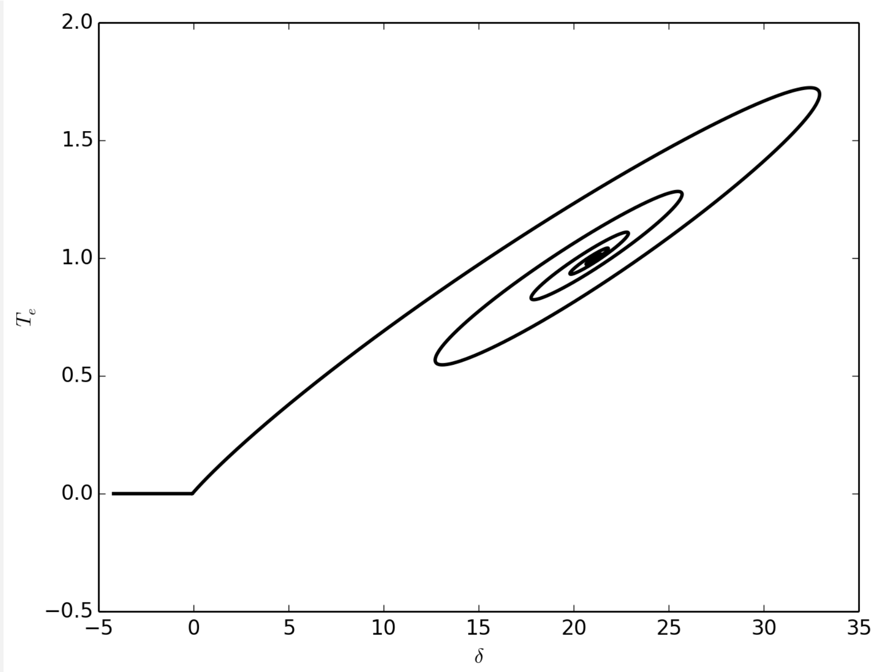


Esta gráfica es muy similar a la de un sistema de segundo orden ante una entrada escalon. De hecho es un caso sub amortiguado en el cual el par eléctrico trata de "seguir" inmediatamente al nuevo par mécanico.

1.1.4 Gráfica ángulo-par

```
plot(maq.data['d'], maq.data['Te'], 'k')
xlabel('$\delta$')
ylabel('$T_e$')
```

```
<matplotlib.text.Text at 0x10aa208d0>
```



Esta gráfica muestra la convergencia del par mecánico y del ángulo hacia un punto de equilibrio común. Esta gráfica es interesante en el sentido que se ve la evolución hacia el estado estable desde un principio.

1.2 Conclusiones

Con este trabajo se corrobora que el modelo fue descrito adecuadamente en un programa de computadora. Así mismo estos resultados son los mismos que están reportados en el libro de Krauze y las gráficas son idénticas. Queda pendiente el modelado de la máquina de inducción en el mismo marco de referencia y posteriormente la inclusión de controles primarios a la máquina síncrona.