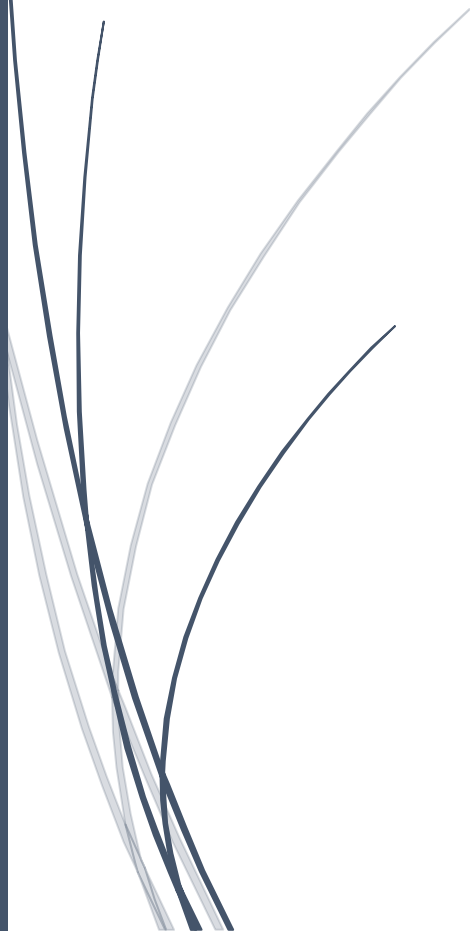


A dark blue vertical bar on the left side of the page. A blue arrow points to the right from the bar, containing the date.

3/26/2020

STR TP N°1

Motor simulado

Several thin, curved lines in dark blue and light grey originate from the bottom left corner and sweep upwards and to the right.

Integrantes:
Brambilla Nicolás
Pereyra Agustín
Perez Sardi Walter

Contents

Contents 1

Introducción al trabajo practico..... 2

Requerimientos 2

Consignas 2

Fecha de entrega: 2

Integrantes:..... 2

Github del proyecto..... 2

Estructura del código..... 2

Manual de usuario..... 3

Resultado obtenido 4

Introducción al trabajo practico

Bienvenidos a sistemas en tiempo real, debido al nuevo virus "Covid-19" se debe realizar la simulación de un motor para graficar sus diferentes comportamientos, tanto estático como dinámico. La idea principal era utilizar el motor físico que se encuentra en el establecimiento IUA pero debido a la cuarentena total no se podrá hacer.

Requerimientos

- a. Respetar las reglas actuales de interacción social, i.e. es preferible colaborar por internet.
- b. En el examen, las presentaciones son individuales. Significa que todos los integrantes deberán estar al tanto de todas las partes del proyecto, no únicamente de su parte.
- c. La presentación deberá incluir un documento que sirve como 'manual de usuario', con información sobre el uso del programa, eventuales problemas, los gráficos producidos, y como repetirlos.

Consignas

1. Implementar una clase 'madre' para unir los tipos de motores
2. Derivar una clase `Simulated_motor`
3. En el program principal (utilizando plantilla `python3_mpl`):
 - a. Instanciar `Simulated_motor`
 - b. Graficar velocidad estática vs torque para obtener el gráfico del PDF
 - c. Graficar comportamiento dinámico (como en el PDF)

Fecha de entrega:

26 de Marzo 2020

Integrantes:

- Brambilla Nicolás
- Pereyra Agustín
- Perez Sardi Walter

Github del proyecto

https://github.com/AguPereyra/str_2020

Estructura del código

La estructura se compone por cuatro archivos los cuales son:

realMotor.py: Aun no posee ningún código porque no lo necesitamos.

simMotor.py: Realizamos el calculo de las velocidades, dinámica y estática para luego aplicar un torque y graficar. Éste cálculo se realiza en dos funciones diferentes porque para el cálculo estático no se debe tener en cuenta el porcentaje de variación de velocidad entre la actual y aquella a llegar (lo que nosotros representamos en la variable *inertia*).

motorTests.py: Definimos los 4 parámetros a ingresar que serán con los cuales se harán los cálculos para las gráficas. Los parámetros, en orden, son: velocidad, sensibilidad, inercia, tiempo entre medidas.

En el caso del grafico dinámico debemos utilizar el torque entre 20 y 50. Además, para éste caso, si hicieramos los cálculos con el motor real, podríamos queres esperar unos segundos

antes de realizar cada medición y modificar la velocidad. Es para ello que existe el 4 parámetro (tiempo entre medidas).

Por otra parte, el parámetro *inertia*, como ya se explicó más arriba, representa el porcentaje de la diferencia entre la velocidad actual y la velocidad a llegar, utilizada en el cálculo dinámico de la velocidad.

Utilizamos la librería matplotlib para realizar las gráficas, indicando con “subplot” las dos graficas.

motor.py: Definimos la interfaz para recibir la velocidad máxima y la sensibilidad en el constructor, como también la función para establecer el torque y otra función para medir la velocidad.

Manual de usuario

Para la utilización del programa, asegurarse que el archivo tenga permisos de ejecución, en caso contrario otorgárselos con el comando “chmod +x motorTests.py”

Una vez obtenidos los permisos ejecutarlo con “./motorTests.py” ingresando contiguamente cuatro parámetros (velocidad, sensibilidad, inercia, tiempo entre medida).

Por ejemplo, utilizamos los siguientes valores que fueron los que nos dio lo más aproximado posible a los gráficos planteados en el PDF:

```
./motorTests.py 255 0.016 15 0
```

Resultado obtenido

