**Título:** Detección de fallas para sistemas de control de procesos continuos

**Hipótesis:** Comparación de métodos de verificación de límites, residuos y machine learning para el TEP

**Plan de trabajo:**

* **Caso Tanque**

Se resolverá la ecuación diferencial de sistema dinámico compuesto por un tanque con nivel inicial Ho y área transversal A, un caudal de entrada Qi y caudal de salida dependiente del nivel H y resistencia hidráulica de válvula de salida R

* + Probar desempeño de funciones caudal, aumento\_gradual\_r y simultank, y medir tiempo de ejecución y exactitud respecto a la solución analítica para distintos pasos de integración
    - caudal: 3 vectores caudal con distintos parámetros
      1. Caudal constante de 24 m3/h, entre 0h y 48h, con ruido 0,01 con paso de 0.1
      2. Caudal constante de 24 m3/h, entre 0h y 48h, con ruido 0,005 con paso de 0.1
      3. Caudal constante de 24 m3/h, entre 0h y 48h, sin ruido con paso de 0.1
    - aumento\_gradual\_r: 3 vectores de resistencia hidráulica con distintos cambios
      1. Resistencia hidráulica de 0h a 48h, que varía de 0,28 a 0.32 entre 1h y 47 con paso de 0.1
      2. Resistencia hidráulica de 0h a 48h, que varía de 0,28 a 0.32 entre 10h y 20h con paso de 0.1
      3. Resistencia hidráulica de 0h a 48h, que varía de 0,28 a 0.32 entre 20,1h y 20,2h con paso de 0.1
    - simultank
      * Comparación entre 5 pasos diferentes (0,0001 ;0,001 ;0,01; 0.1; 1) respecto a solución analítica entre 0h y 10h
      * 3 vectores de nivel para las combinaciones anteriores 1-1, 2-2 y 3-3, área 3m2.
  + Simular distintas fallas

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | **Componente/Causa** | | |
| **Tanque** | **Tipo de Falla** | RH | Fuga | Sensor |
| Drift | x |  |  |
| Step | x | x | x |
| Varianza |  |  | x |
|  | Pulse |  |  | x |

* + - * Para cada falla se variará la amplitud de la falla y su duración en el caso Drift y su frecuencia en el caso de Corta para el Sensor
  + Aplicar métodos de detección de fallas
    - Verificación de límites
      * Valores absolutos
        + (Hmín < H < Hmáx)
      * Tendencia
      * Cambio de media
        + 2 Niveles de confiabilidad (0,95 y 0,99)
        + 2 Δμ (0,01 y 0,005)
      * Cambio de varianza
        + 2 Niveles de confiabilidad (0,95 y 0,99)
        + 2 Δσ (Por definir)
  + Calcular los índices de desempeño
    - FDR: False detection rate
    - FAR: False alarm rate
    - TDF: Time to detect fault

**Metodología**

Se trabajará con Python 3.6, específicamente en el IDE PyCharm Community Edition 2016.3.3.

Se creó una función llamada “simultank” de la forma siguiente (Corregir todas las funciones con el código final nombre función):

*"""  
Creamos una funcion llamada simultank que resuelve la ecuacion diferencial que describe el comportamiento  
de la altura del tanque en funcion del tiempo.  
  
Los parametros de entrada de nuestra funcion son:  
  
area: Area transversal del tanque  
nivel\_inicial: Altura inicial del contenido del tanque  
resist\_hidraulica: Resistencia hidraulica de la valvula de salida  
caudal\_entrada: Caudal de entrada  
tiempo\_inicial: Tiempo inicial desde el cual se quiere simular el comportamiento  
tiempo\_final: Tiempo final desde el cual se quiere simular el comportamiento  
paso: El incremento de tiempo entre ti y tf  
  
Los parametros de salida son:  
vector\_H: vector de alturas H en funcion del tiempo  
"""***import** numpy **as** np  
**def simultank**(area, nivel\_inicial, resist\_hidraulica, caudal\_entrada, tiempo\_inicial, tiempo\_final, paso):  
  
  
 vector\_tiempo = np.arange(tiempo\_inicial, tiempo\_final, paso) # Generamos el vector de intervalos de tiempo  
 vector\_H = np.array([nivel\_inicial]) # Creamos el vector de altura que estara en funcion del tiempo  
 vector\_K1 = np.array([0]) # K1 y K2 son los parametros del metodo RK2  
 vector\_K2 = np.array([0])  
  
 iteraciones = len(vector\_tiempo)  
 contador = 0  
  
 **while** contador < iteraciones - 1: # Este ciclo ira iterando para cada valor de tiempo  
  
 ''' Aplicamos RK2  
 '''  
 vector\_K1\_mas\_1 = paso\*((caudal\_entrada[contador] / area) - (nivel\_inicial / (area \* resist\_hidraulica[contador])))  
 vector\_K2\_mas\_1 = paso\*((caudal\_entrada[contador] / area) - (nivel\_inicial + vector\_H[contador] + vector\_K1[contador]) / (area \* resist\_hidraulica[contador]))  
 H\_mas\_1 = vector\_H[contador] + 0.5\*(vector\_K1[contador] + vector\_K2[contador])  
  
 contador += 1  
  
 vector\_H = np.append(vector\_H, H\_mas\_1) # El vector de altura va creciendo para cada t  
 vector\_K1 = np.append(vector\_K1,vector\_K1\_mas\_1 )  
 vector\_K2 = np.append(vector\_K2, vector\_K2\_mas\_1)  
  
 **return** vector\_H

Se creó una función llamada “caudal” de la forma siguiente:

*"""  
Programa que genera un vector de caudales  
"""*'''  
  
Parametros de entrada  
  
 set\_point: Valor que tendra el caudal a lo largo del vector  
 longitud: Longitud del vector  
 paso: Paso de integracion  
 factpr\_ruido: Amplitud del ruido gausiano  
 ruido: De ser True, el vector tendra ruido gausiano.  
  
Parametros de salida  
  
 Q: Vector de caudales  
  
'''  
  
  
**def caudal**(set\_point, longitud, paso, factor\_ruido=0.01, ruido=**False**):  
  
 **import** numpy **as** np  
 **if** int(longitud/paso) - longitud/paso == 0:  
 Q = np.ones(longitud/paso)\*set\_point  
  
 **if** ruido:  
 np.random.seed(0)  
 Q += set\_point\*factor\_ruido\*np.random.normal(0, 1, longitud/paso)  
  
 **else**:  
 print('longitud/paso debe ser un entero')  
 Q=[]  
  
 **return** Q

Se creó una función llamada “aumento\_gradual\_r” de la forma siguiente

*"""  
Funcion que genera un vector de valores para resistencia hidraulica de una valvula con un incremento de r\_inicial a  
r\_final desde un tiempo inicial t\_inicial hasta un tiempo final t\_final  
"""*'''  
  
Parametros de entrada  
  
 r\_inicial: Valor inicial de la resistencia hidraulica  
 r\_final: Valor final de la resistencia hidraulica  
 longitud: Longitud del vector  
 t\_inicial: Tiempo en que inicia el cambio gradual  
 t\_final: Tiempo en que finaliza el cambio gradual  
 paso: Paso de integracion  
  
Parametros de salida  
  
 R: vector de resistencias hidraulicas  
  
'''  
  
  
**def aumento\_gradual\_r**(r\_inicial, r\_final, longitud, t\_inicial, t\_final, paso):  
  
 **import** numpy **as** np  
  
 **if** int(longitud/paso) - longitud/paso == 0:  
 R = np.ones(longitud/paso)\*r\_inicial  
 R[t\_inicial/paso:t\_final/paso] = np.arange(r\_inicial, r\_final, (r\_final-r\_inicial)/((t\_final-t\_inicial)/paso))  
 R[t\_final/paso:] = np.ones(longitud/paso-t\_final/paso)\*r\_final  
 **else**:  
 print('longitud/paso debe ser un entero')  
 R=[]  
  
 **return** R

**Resultados**

* **Evaluación del desempeño de las funciones**

Las variables para la evaluación de las funciones fueron las siguientes:

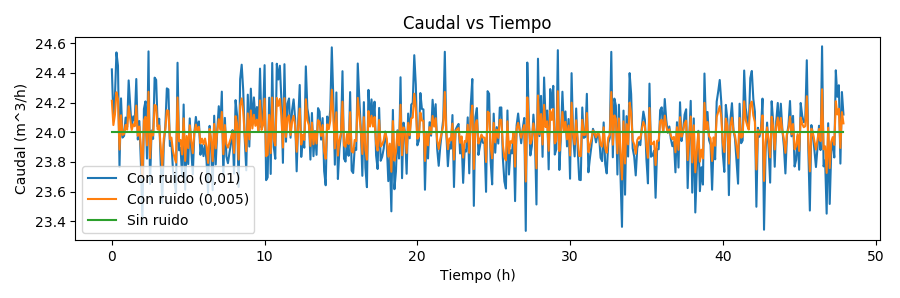
r\_inicial = 0.28 # Valor inicial de la resistencia hidraulica  
r\_final = 0.32 # Valor final de la resistencia hidraulica  
longitud = 48 # Tiempo de simulacion [h]  
t\_inicial\_falla\_1 = 1 # Tiempo en que inicial la falla [h]  
t\_final\_falla\_1 = 47 # Tiempo en que se estabiliza la falla [h]  
t\_inicial\_falla\_2 = 10 # Tiempo en que inicial la falla [h]  
t\_final\_falla\_2 = 20 # Tiempo en que se estabiliza la falla [h]  
t\_inicial\_falla\_3 = 20.1 # Tiempo en que inicial la falla [h]  
t\_final\_falla\_3 = 20.3 # Tiempo en que se estabiliza la falla [h]  
paso = 0.1 # Paso de integracion [h]  
q\_set = 24 # Caudal de entrada [m^3/h]  
ruido = **True** # Ruido en el caudal de entrada  
area = 2 # Area transversal del tanque [m^2]  
nivel\_inicial = 0 # Nivel inicial del tanque [m]  
t\_inicial = 0 # Tiempo inciial de la simulacion [h]  
t\_final = longitud # Tiempo final de la simulacion [h]  
conf\_lev = 0.99 # Nivel de confiabilidad  
tiempo = np.arange(t\_inicial,t\_final,paso) # Vector de rangos de tiempo de la simulacion

* + **caudal**

Al aplicar la función y graficar, se obtuvo lo siguiente

Qi\_ruido\_0\_01 = caudal(set\_point=q\_set, longitud=longitud, paso=paso, factor\_ruido=0.01, ruido=**True**)  
Qi\_ruido\_0\_005 = caudal(set\_point=q\_set, longitud=longitud, paso=paso, factor\_ruido=0.005, ruido=**True**)  
Qi\_no\_ruido = caudal(set\_point=24, longitud=longitud, paso=paso, ruido=**False**)  
plt.figure(figsize=(9, 3))  
plt.plot(tiempo,Qi\_ruido\_0\_01,label='Con ruido (0,01)')  
plt.plot(tiempo,Qi\_ruido\_0\_005,label='Con ruido (0,005)')  
plt.plot(tiempo,Qi\_no\_ruido,label='Sin ruido')  
plt.title('Caudal vs Tiempo')  
plt.ylabel('Caudal (m^3/h)')  
plt.xlabel('Tiempo (h)')  
plt.legend()  
plt.tight\_layout()

plt.show()

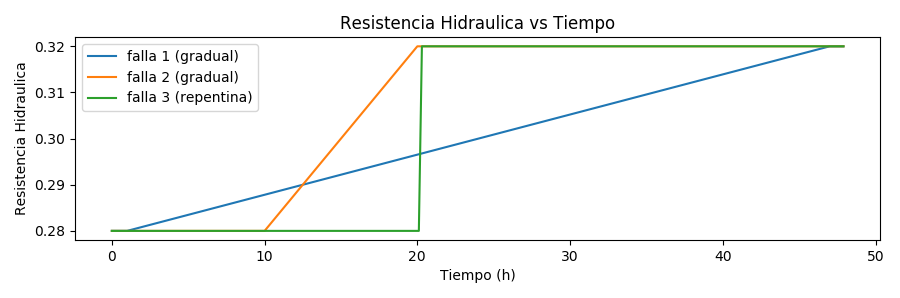
****

* + **aumento\_gradual\_r**

Al aplicar la función y graficar, se obtuvo lo siguiente

# aumento\_gradual\_r  
r\_1 = aumento\_gradual\_r(r\_inicial=r\_inicial, r\_final=r\_final, longitud=longitud,   
 t\_inicial=t\_inicial\_falla\_1, t\_final=t\_final\_falla\_1, paso=paso)  
r\_2 = aumento\_gradual\_r(r\_inicial=r\_inicial, r\_final=r\_final, longitud=longitud,   
 t\_inicial=t\_inicial\_falla\_2, t\_final=t\_final\_falla\_2, paso=paso)  
r\_3 = aumento\_gradual\_r(r\_inicial=r\_inicial, r\_final=r\_final, longitud=longitud,   
 t\_inicial=t\_inicial\_falla\_3, t\_final=t\_final\_falla\_3, paso=paso)  
plt.figure(figsize=(9, 3))  
plt.plot(tiempo,r\_1,label='falla 1 (gradual)')  
plt.plot(tiempo,r\_2,label='falla 2 (gradual)')  
plt.plot(tiempo,r\_3,label='falla 3 (repentina)')  
plt.title('Resistencia Hidraulica vs Tiempo')  
plt.ylabel('Resistencia Hidraulica')  
plt.xlabel('Tiempo (h)')  
plt.legend()  
plt.tight\_layout()

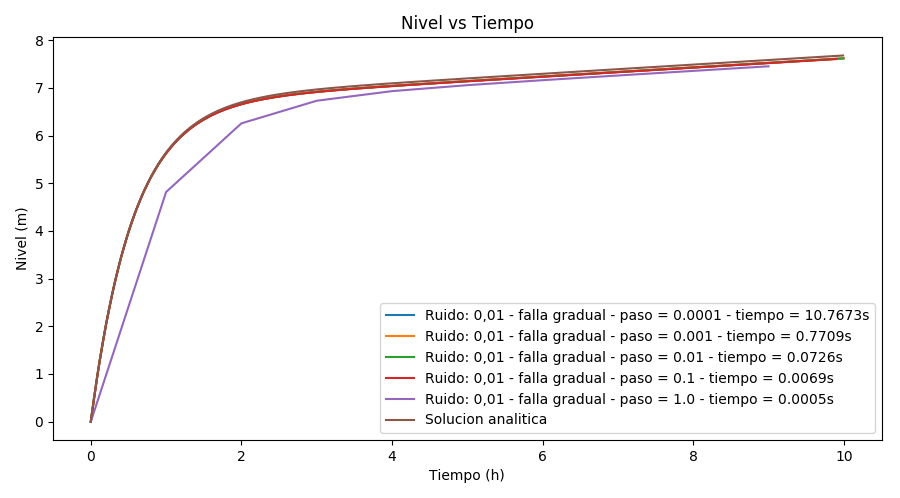
plt.show()



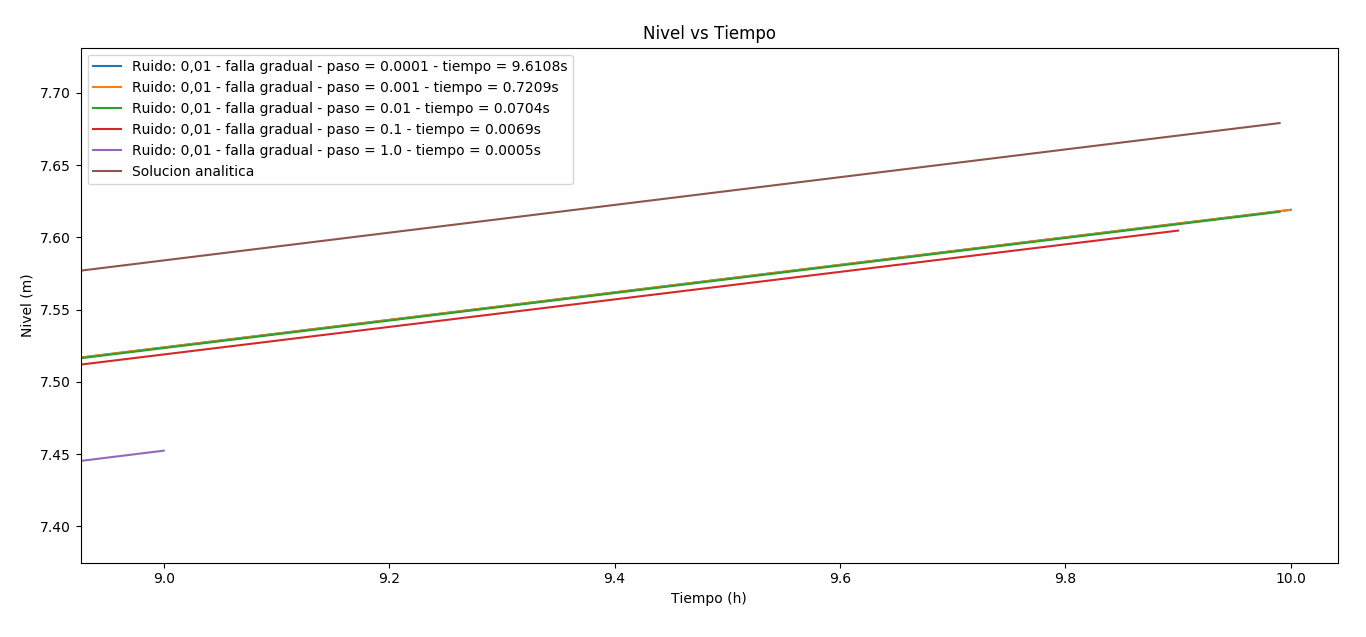
* + **simultank**

Para comprar distintos pasos de integración, se usó el siguiente código, obteniendo la gráfica a continuación

plt.figure(figsize=(9, 5))  
**for** paso **in** np.logspace(-4,0,5,endpoint=**True**):  
 start = timer()  
 Qi = caudal(set\_point=q\_set, longitud=10, paso=paso, ruido=**False**)  
 r = aumento\_gradual\_r(r\_inicial=r\_inicial, r\_final=r\_final, longitud=10, t\_inicial=0, t\_final=10, paso=paso)  
 nivel = simultank(area=area, nivel\_inicial=nivel\_inicial, resist\_hidraulica=r, caudal\_entrada=Qi,  
 tiempo\_inicial=0, tiempo\_final=10, paso=paso,analitic\_sol=**False**)  
 end = timer()  
 tiempo = end - start  
 plt.plot(np.arange(0,10,paso),nivel, label='Ruido: 0,01 - '  
 'falla gradual - paso = {} - tiempo = {:.4f}s'.format(paso,tiempo))  
Qi\_analitico = caudal(set\_point=q\_set, longitud=10, paso=0.01, ruido=**False**)  
r\_analitico = aumento\_gradual\_r(r\_inicial=r\_inicial, r\_final=r\_final, longitud=10, t\_inicial=0, t\_final=10, paso=0.01)  
nivel\_analitico = simultank(area=area, nivel\_inicial=nivel\_inicial, resist\_hidraulica=r\_analitico,  
 caudal\_entrada=Qi\_analitico,tiempo\_inicial=0, tiempo\_final=10, paso=0.01,analitic\_sol=**True**)  
plt.plot(np.arange(0,10,0.01),nivel\_analitico,label='Solucion analitica')  
plt.title('Nivel vs Tiempo')  
plt.ylabel('Nivel (m)')  
plt.xlabel('Tiempo (h)')  
plt.legend()  
plt.tight\_layout()  
plt.show()



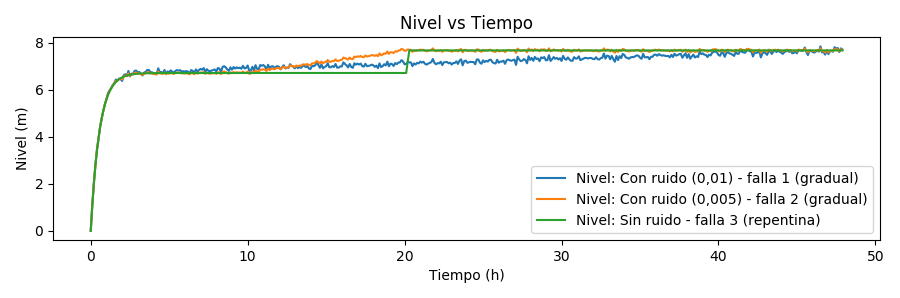
Si se agranda el área final, cerca de las 10h, puede apreciarse la diferencia entre los pasos elegidos:



Al aplicar la función y graficar, se obtuvo lo siguiente

# simultank  
nivel\_1 = simultank(area=area, nivel\_inicial=nivel\_inicial, resist\_hidraulica=r\_1, caudal\_entrada=Qi\_ruido\_0\_01,  
 tiempo\_inicial=t\_inicial, tiempo\_final=t\_final, paso=paso, analitic\_sol=**False**)  
nivel\_2 = simultank(area=area, nivel\_inicial=nivel\_inicial, resist\_hidraulica=r\_2, caudal\_entrada=Qi\_ruido\_0\_005,  
 tiempo\_inicial=t\_inicial, tiempo\_final=t\_final, paso=paso, analitic\_sol=**False**)  
nivel\_3 = simultank(area=area, nivel\_inicial=nivel\_inicial, resist\_hidraulica=r\_3, caudal\_entrada=Qi\_no\_ruido,  
 tiempo\_inicial=t\_inicial, tiempo\_final=t\_final, paso=paso, analitic\_sol=**False**)  
plt.figure(figsize=(9, 3))  
plt.plot(tiempo,nivel\_1,label='Nivel: Con ruido (0,01) - falla 1 (gradual)')  
plt.plot(tiempo,nivel\_2,label='Nivel: Con ruido (0,005) - falla 2 (gradual)')  
plt.plot(tiempo,nivel\_3,label='Nivel: Sin ruido - falla 3 (repentina)')  
plt.title('Nivel vs Tiempo')  
plt.ylabel('Nivel (m)')  
plt.xlabel('Tiempo (h)')  
plt.legend()  
plt.tight\_layout()

plt.show()



* **Caso Tanque**
  + Simular distintas fallas
    - Aumento gradual de resistencia hidráulica en la válvula de salida
      * Aumento pequeño (0,28:0,45) en 4 períodos (1 año, 2 años, 3 años y 4 años)
    - Aumento súbito de resistencia hidráulica en la válvula de salida
      * 4 Aumentos diferentes (0,28: 0,3; 0,4; 0,6; 0,8)
    - Falla en el sensor de nivel: no reporta nivel