03_Analyze_Indexed_Files

March 25, 2020

1 Basic MDF processing.

[1]:
```
from mdf_models import Channel, MDF, db

from pony.orm import *
```

TODO: Make this more descriptive for notebooks.

[2]:
```
# Select the first MDF file
q = MDF.select()
# Get the first MDF indexed.
mdf_sql = q.first()
mdf_sql
```

[2]:
```
MDF[1]
```

2 List the channels in the file.

TODO: Make *this* more descriptive.

[3]:
```
list(mdf_sql.channels)
```

[3]:
```
[Channel[3],
 Channel[6],
 Channel[10],
 Channel[5],
 Channel[7],
 Channel[2],
 Channel[12],
 Channel[4],
 Channel[1],
 Channel[11],
 Channel[9],
 Channel[8]]
```

[4]:
```
# For each of the channels in the MDF file:
for channel in mdf_sql.channels:
```

```
    # Print the channel name
    print(channel.name)
```

```
engine_speed_desired
coolant_temp
efficiency
transmission_gear
longitude
engine_speed
Y
vehicle_speed
time
X
power
latitude
```

[5]:
```
channel.name
```

[5]:
```
'latitude'
```

[6]:
```
channel
```

[6]:
```
Channel[8]
```

[7]:
```
# Set of MDFs that have the channel above.
channel.mdfs
```

[7]:
```
<MDFSet Channel[8].mdfs>
```

Get the channel for engine speed.

[8]:
```
engine_speed = Channel.select().filter(lambda channel: channel.name ==␣
↪"engine_speed").first()
```

[9]:
```
engine_speed
```

[9]:
```
Channel[2]
```

3 Find MDFs missing channel

MDFs missing engine speed:

[10]:
```
bad_mdfs = list()
for mdf in MDF.select():
    if channel not in mdf.channels:
        bad_mdfs.append(mdf)
```

[11]:
```
for bad_mdf in bad_mdfs:
    break
```

[12]:
```
bad_mdf.path
```

[12]:
```
'/projects/MDF_Data_Pipeline/Data/CarCompanyLLC/Boat/7218f46b-2da1-4722-a659-96b
4bb197313.mdf'
```

[13]:
```
bad_mdf.size_mb
```

[13]:
```
20.00023365020752
```

[14]:
```
bad_mdf.path
```

[14]:
```
'/projects/MDF_Data_Pipeline/Data/CarCompanyLLC/Boat/7218f46b-2da1-4722-a659-96b
4bb197313.mdf'
```

4 Find MDF Files By Size

Big MDFs.

[15]:
```
query = select(mdf for mdf in MDF
               if mdf.size_mb>1024)
query.count()
```

[15]:
```
3
```

Medium MDFs.

[16]:
```
query = select(mdf for mdf in MDF
               if mdf.size_mb<100 and mdf.size_mb>50)
query.count()
```

[16]:
```
295
```

Small MDFs.

[17]:
```
query = select(mdf for mdf in MDF
               if mdf.size_mb<1)
query.count()
```

[17]:
```
1000
```

[18]:
```
mdf_obj = query.first()
mdf_obj
```

[18]:
```
MDF[1]
```

[19]:
```
mdf_obj.path
```

[19]:
```
'/projects/MDF_Data_Pipeline/Data/HeavyEquipmentInc/Airplane/5ab19863-6324-41b4-
a7c0-387a38d00d3c.mf4'
```

[20]:
```
mdf_obj.size
```

[20]:
```
171256.0
```

[21]:
```
mdf_obj.size_mb
```

[21]:
```
0.16332244873046875
```

[22]:
```
MDF
```

[22]:
```
mdf_models.MDF
```

#namespaces.

[23]:
```
import asammdf
```

[24]:
```
mdf = asammdf.MDF(mdf_obj.path)
```

[25]:
```
asammdf.__version__
```

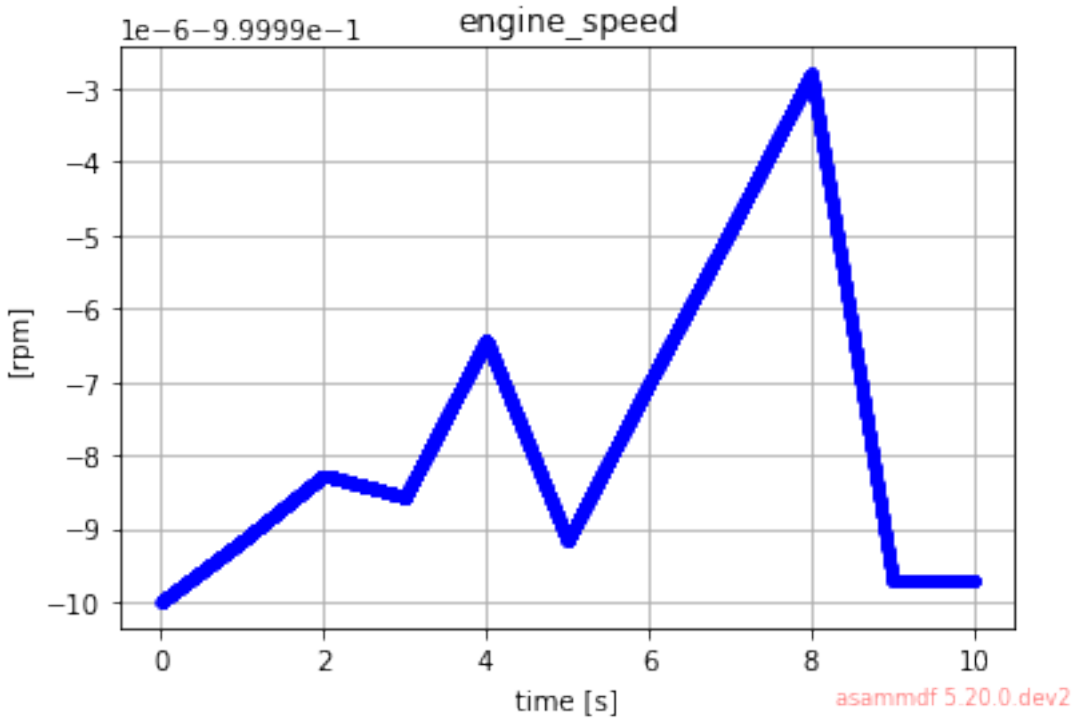
[25]:
```
'5.20.0.dev2'
```

[26]:
```
mdf.channels_db
```

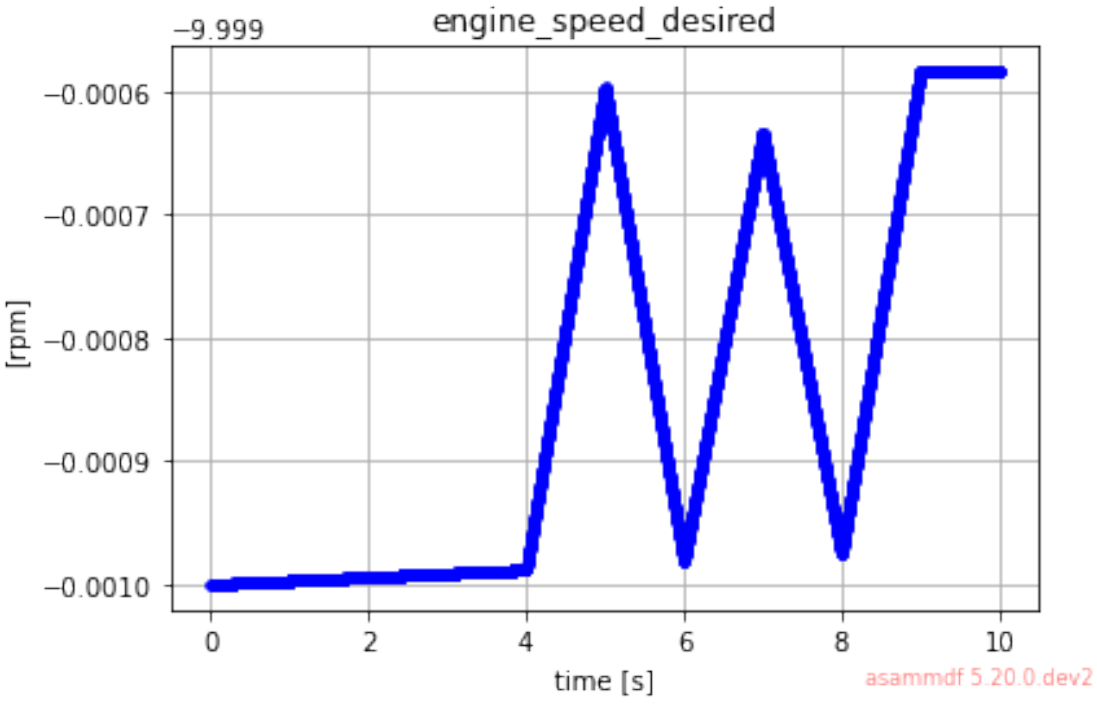
[26]:
```
{'time': ((0, 0),),
 'engine_speed': ((0, 1),),
 'engine_speed_desired': ((0, 2),),
 'vehicle_speed': ((0, 3),),
 'transmission_gear': ((0, 4),),
 'coolant_temp': ((0, 5),),
 'longitude': ((0, 6),),
 'latitude': ((0, 7),),
 'power': ((0, 8),),
 'efficiency': ((0, 9),),
 'X': ((0, 10),),
 'Y': ((0, 11),)}
```

[27]:
```
for channel in mdf.iter_channels():
    channel.plot()
```

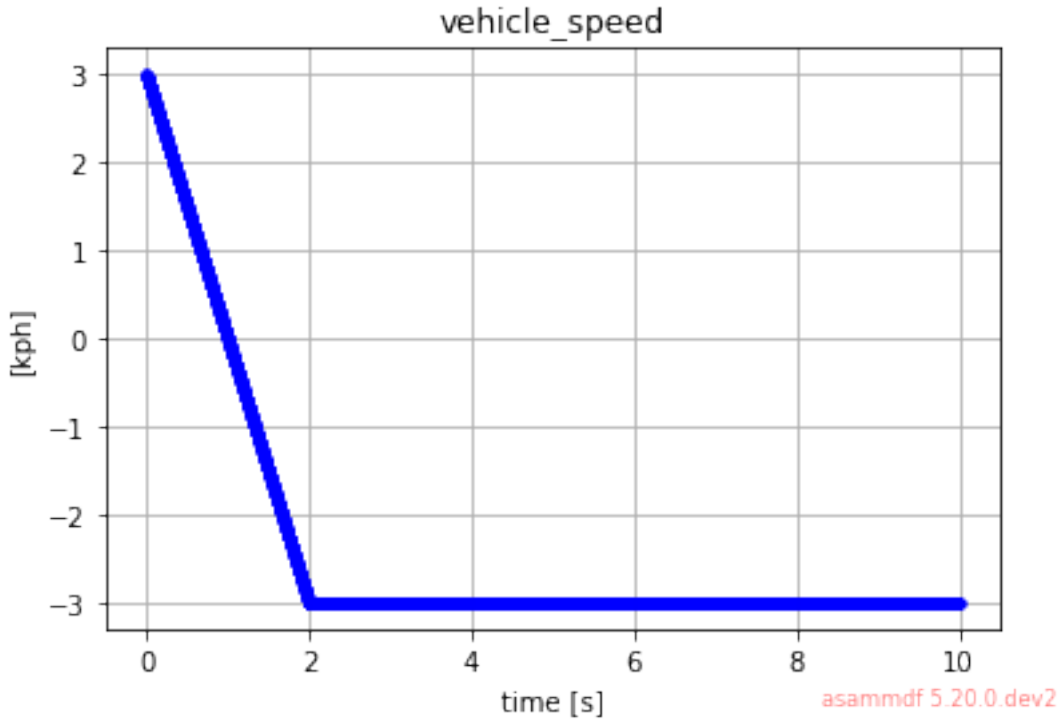
```
WARNING:root:Signal plotting requires pyqtgraph or matplotlib
```



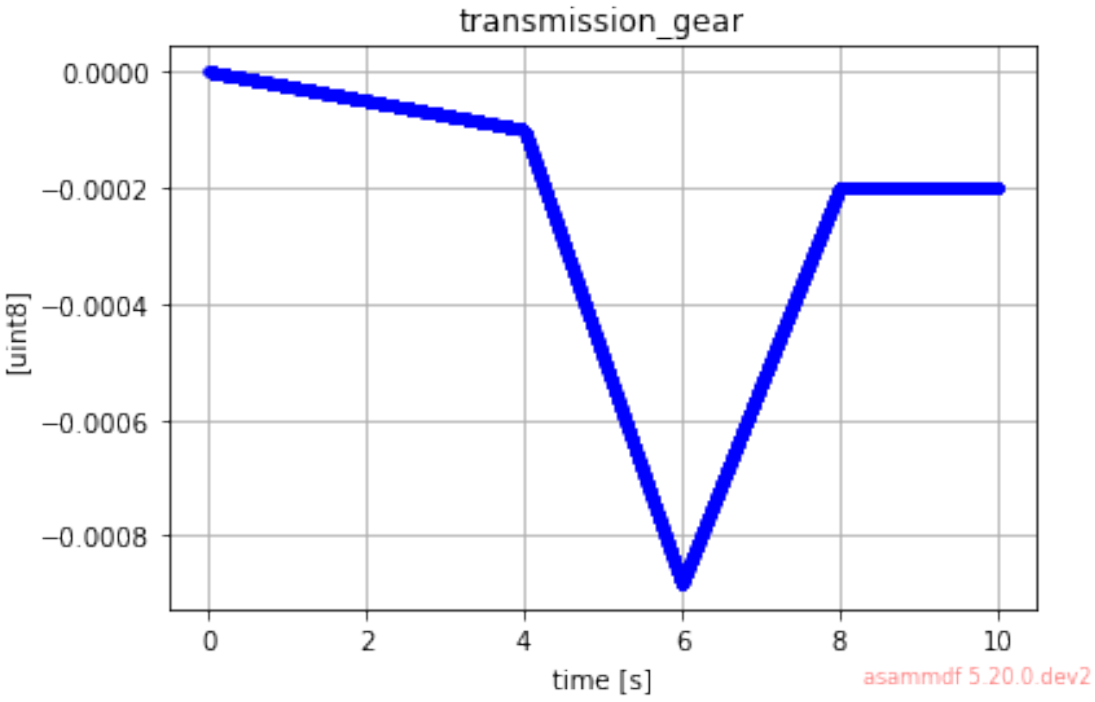

```
WARNING:root:Signal plotting requires pyqtgraph or matplotlib
```



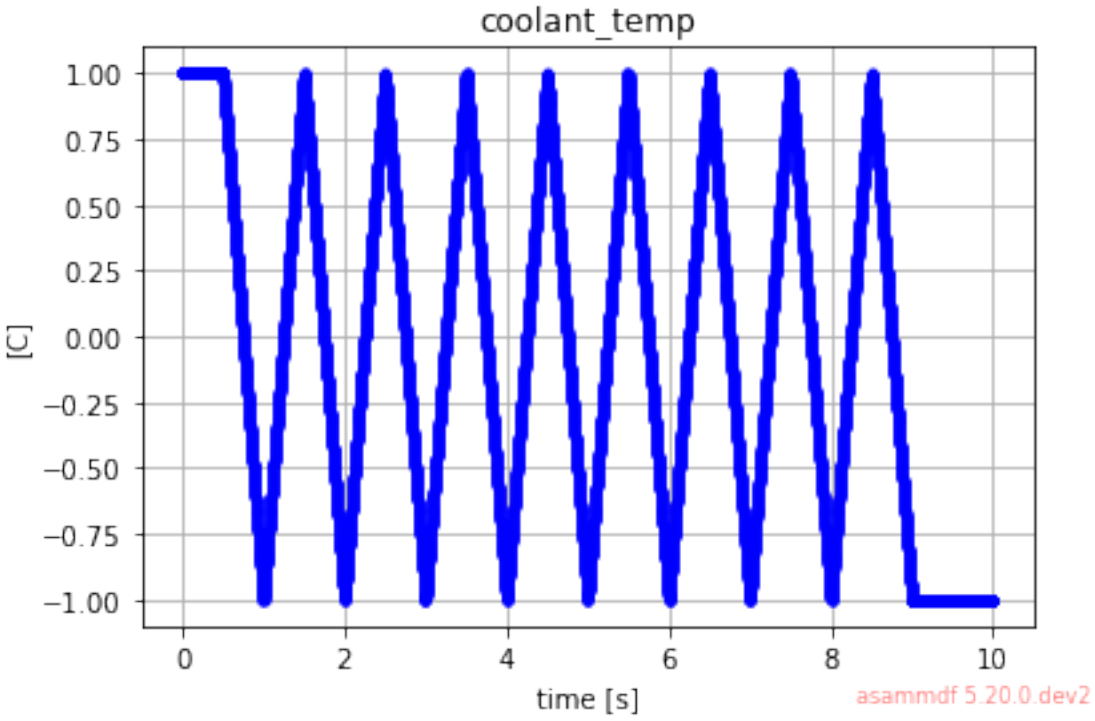
```
WARNING:root:Signal plotting requires pyqtgraph or matplotlib
```



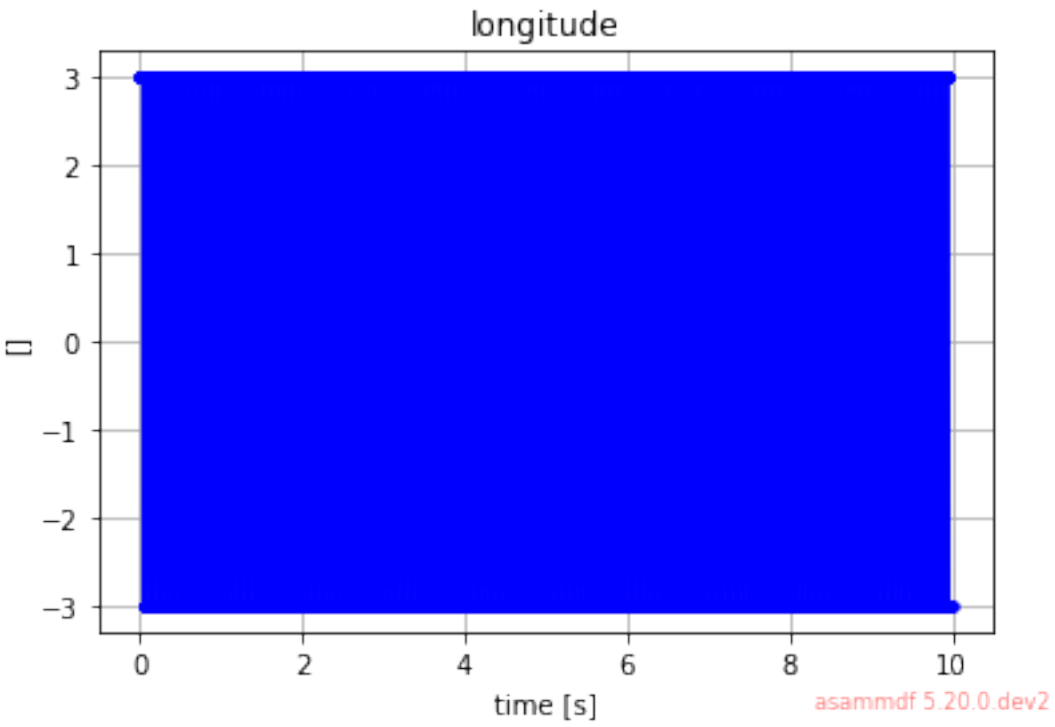

```
WARNING:root:Signal plotting requires pyqtgraph or matplotlib
```



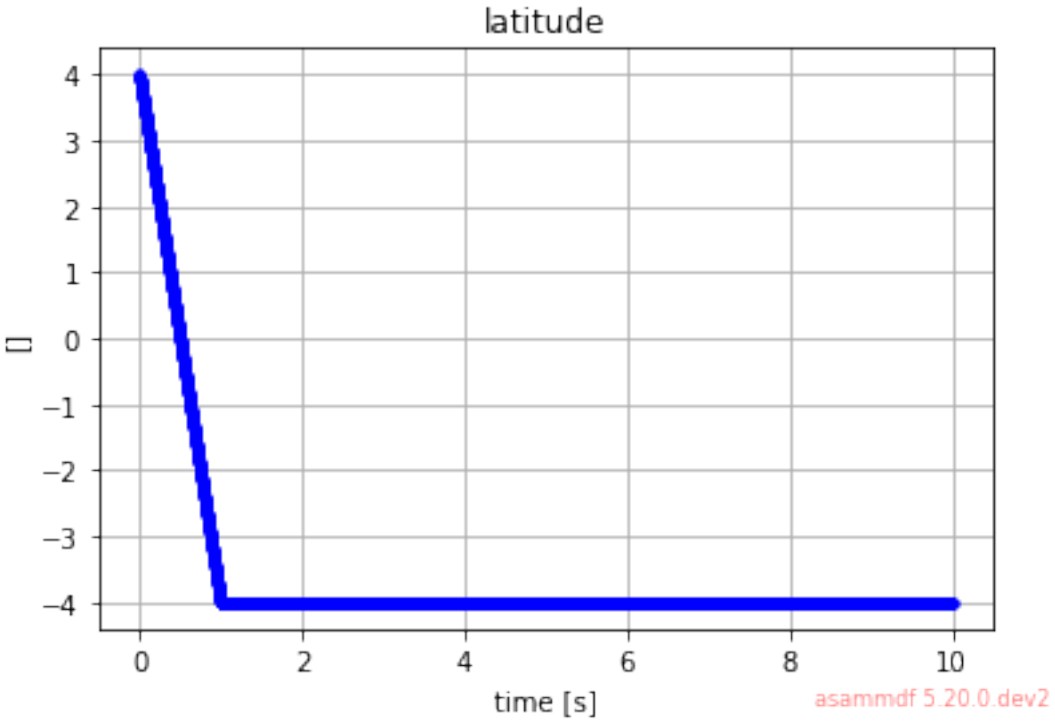
```
WARNING:root:Signal plotting requires pyqtgraph or matplotlib
```



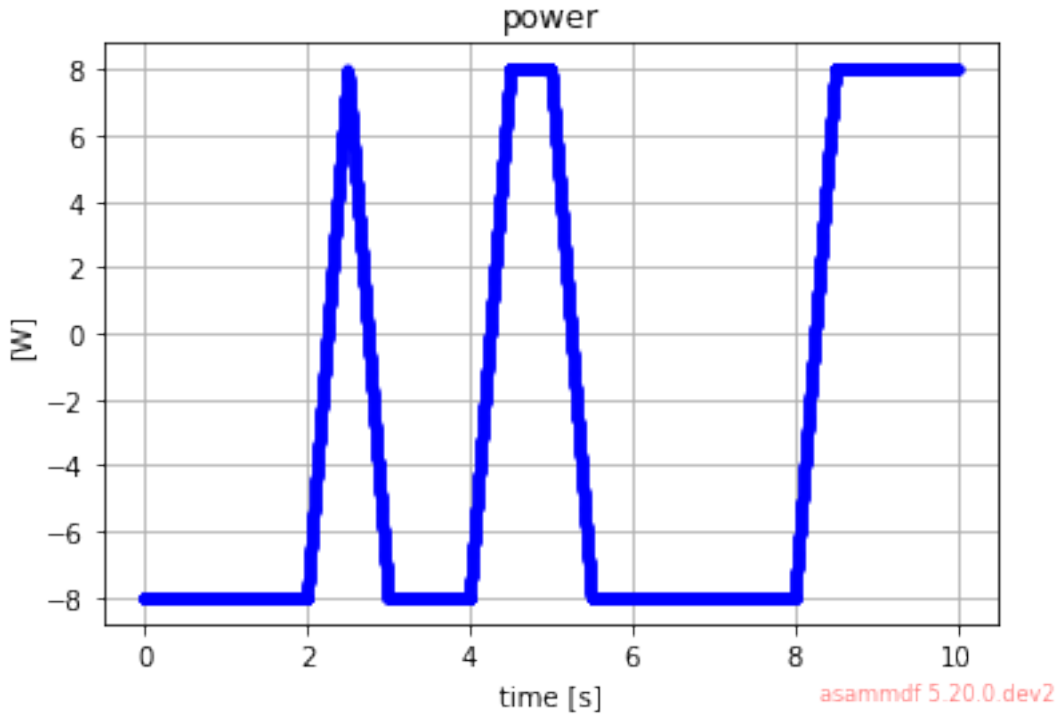

```
WARNING:root:Signal plotting requires pyqtgraph or matplotlib
```



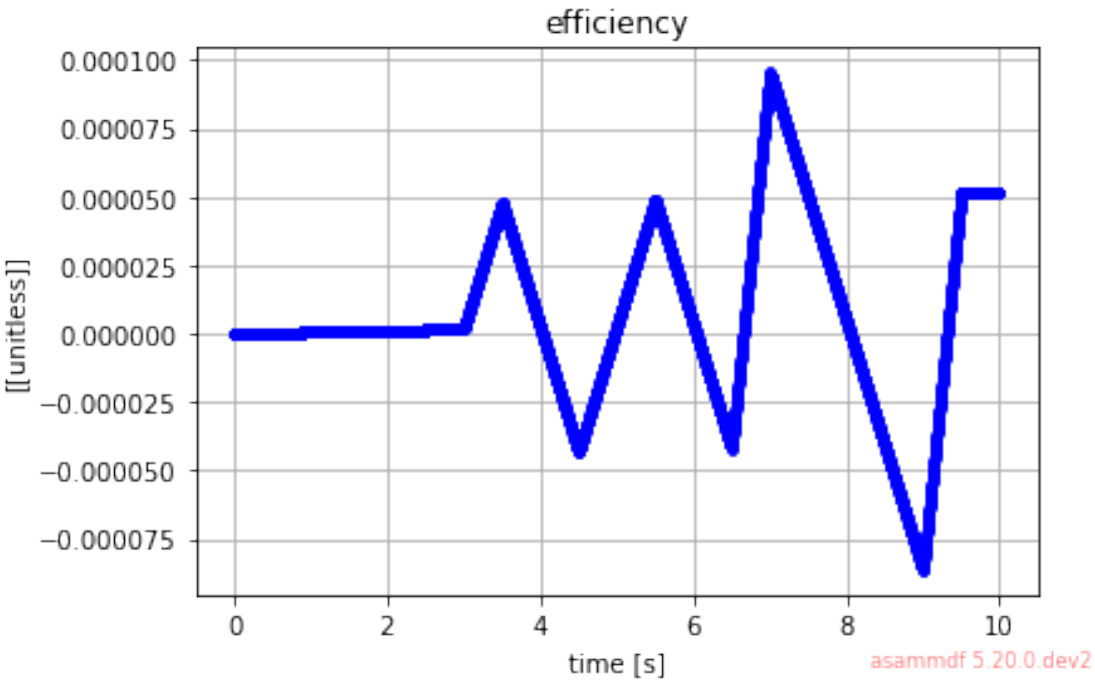

```
WARNING:root:Signal plotting requires pyqtgraph or matplotlib
```



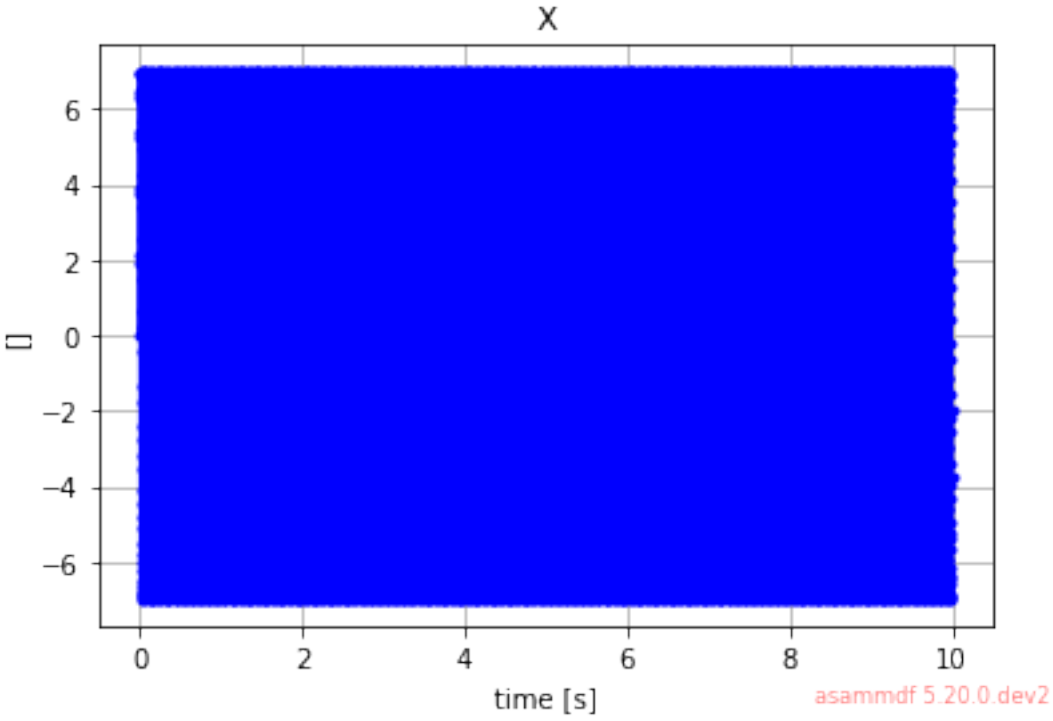
```
WARNING:root:Signal plotting requires pyqtgraph or matplotlib
```



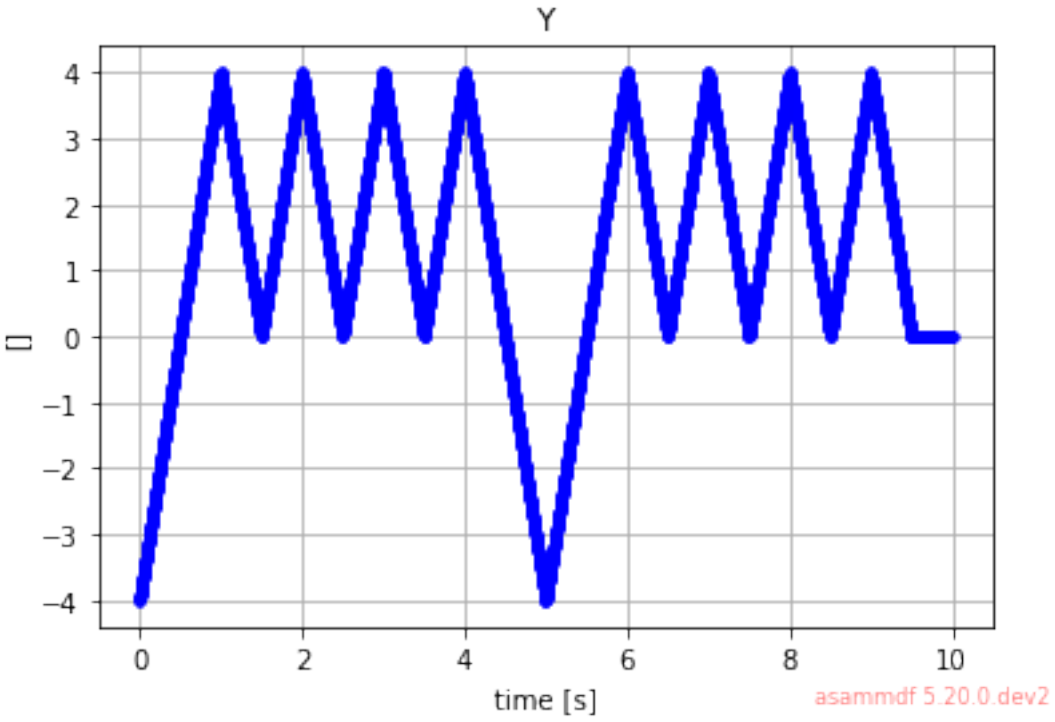

```
WARNING:root:Signal plotting requires pyqtgraph or matplotlib
```




```
WARNING:root:Signal plotting requires pyqtgraph or matplotlib
```



```
WARNING:root:Signal plotting requires pyqtgraph or matplotlib
```



[28]:
```
import numpy as np
```

[29]:
```
np.mean(np.diff(channel.timestamps))
```

[29]:
```
0.0009992006145014896
```