Analyse vidéo

March 16, 2024

[ ]:

```
import cv2
from matplotlib import pyplot as plt
import numpy as np
import ipywidgets as widgets
import scienceplots
import scipy as sp
import pandas as pd

plt.style.use(['science','notebook','grid'])
```

[ ]:

```
m40 = 40e-3
m60 = 60e-3
m70 = 70e-3
m80 = 80e-3
m100 = 100e-3

r_p = 1.5e-2
r_m = 3e-2
r_g = 4.5e-2
```

[ ]:

```
quad = lambda x, a, b, c: a*x**2 + b*x + c
lin = lambda x, a, b: a*x + b
```

1 Analyse vidéos

1.1 Vidéo m40

[ ]:

```
video_path = "m40.mp4"
cap = cv2.VideoCapture(video_path)
```

[ ]:

```
frame_rate = cap.get(cv2.CAP_PROP_FPS)
print("Frame rate: ", frame_rate)
```

```
Frame rate:  59.78801512879166
```

[ ]:

```
frames = []
while True:
    ret, frame = cap.read()
    if not ret:
        break
    frames.append(frame)
```

[ ]:
```
frame_rate = cap.get(cv2.CAP_PROP_FPS)
print("Frame rate: ", frame_rate)
```

```
Frame rate:  59.78801512879166
```

[ ]:
```
def disp_f(num):
    plt.imshow(frames[num])

widgets.interact(disp_f, num=widgets.IntSlider(min=0, max=len(frames)-1,␣
 ↪step=1))
```

```
interactive(children=(IntSlider(value=0, description='num', max=1183),␣
 ↪Output()), _dom_classes=('widget-intera…
```

[ ]:
```
<function __main__.disp_f(num)>
```

[ ]:
```
analyse = frames[363:1098]
```

[ ]:
```
def pos_track(x,y):
    image = analyse[0].copy()
    w, h = 50, 50
    cv2.rectangle(image, (x, y), (x + w, y + h), (255, 0, 0), 4)
    plt.imshow(image)

widgets.interact(pos_track, x=widgets.IntSlider(min=0, max=len(analyse[0]),␣
 ↪step=1), y=widgets.IntSlider(min=0, max=len(analyse[0][0]), step=1))
```

```
interactive(children=(IntSlider(value=0, description='x', max=1080),␣
 ↪IntSlider(value=0, description='y', max=1…
```

[ ]:
```
<function __main__.pos_track(x, y)>
```

[ ]:
```
tracker = cv2.TrackerCSRT_create()

track_frames = analyse.copy()

init_trac_f = track_frames[0]
roi = (357, 372, 50, 50)

tracker.init(init_trac_f, roi)

results = []
frame_tracked = []
```

```
for i in range(1, len(track_frames)):
    # Read the next frame
    frame = track_frames[i]

    # Update the tracker
    success, roi = tracker.update(frame)

    if success:
        # Draw a bounding box around the tracked object
        (x, y, w, h) = tuple(map(int, roi))
        results += [(x,y,w,h)]
        cv2.rectangle(frame, (x, y), (x + w, y + h), (255, 0, 0), 4)
        frame_tracked.append(frame)
```

[ ]:

```
def disp_t(num):
    plt.imshow(frame_tracked[num])

widgets.interact(disp_t, num=widgets.IntSlider(min=0, max=len(frame_tracked)-1,␣
 ↪step=1))
```

```
interactive(children=(IntSlider(value=0, description='num', max=733), Output()),␣
 ↪_dom_classes=('widget-interac…
```

[ ]: `<function __main__.disp_t(num)>`

[ ]:

```
good_results = results
```

[ ]:

```
center_of_rot = (705, 535, 50, 50)
```

[ ]:

```
def pos_from_case(case):
    x,y,w,h = case
    return (x + (x+w))/2, (y + (y+h))/2
```

[ ]:

```
pos_c = pos_from_case(center_of_rot)
pos_r = [pos_from_case(i) for i in good_results]
```

[ ]:

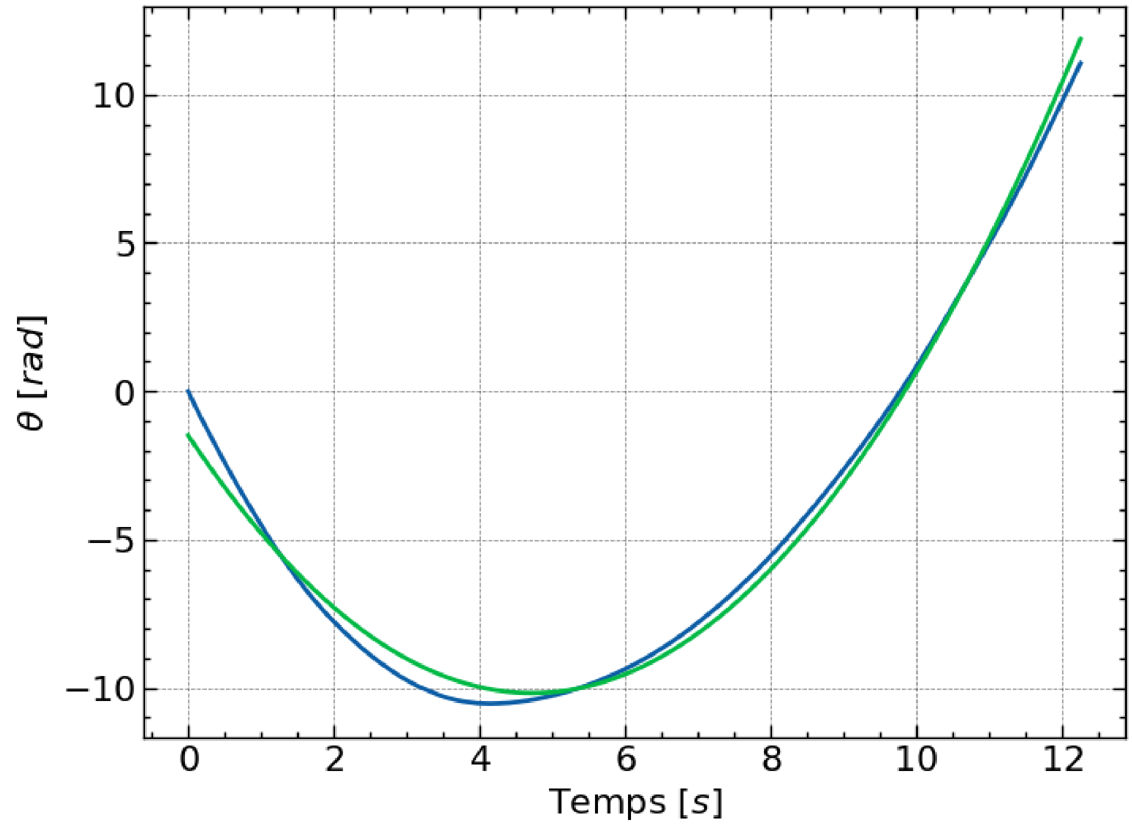
```
theta = []
cur_theta = 0

prev_dir = [pos_r[0][0]-pos_c[0], pos_r[0][1]-pos_c[1]]
for i in range(len(pos_r)):
    next_dir = [pos_r[i][0]-pos_c[0], pos_r[i][1]-pos_c[1]]
    cur_theta += (np.arctan2(next_dir[1], next_dir[0]) - np.
 ↪arctan2(prev_dir[1], prev_dir[0]))
    if np.abs(np.arctan2(next_dir[1], next_dir[0]) - np.arctan2(prev_dir[1],␣
 ↪prev_dir[0])) > np.pi:
        cur_theta -= 2*np.pi*np.sign(np.arctan2(next_dir[1], next_dir[0]) - np.
 ↪arctan2(prev_dir[1], prev_dir[0]))
    theta.append(cur_theta)
    prev_dir = next_dir

print(theta)
```

```
[0.0, -0.08778354760691487, -0.17059950754332487, -0.25515083478660117,
-0.3420496079415507, -0.42853843111429235, -0.5163805436501132,
-0.5956304505778518, -0.6803220071901923, -0.7691987958307247,
-0.8501164449283758, -0.9292067548102327, -1.0118297128772062,
-1.0895543483739951, -1.1632919784526496, -1.2438661421891992,
-1.3317588827435198, -1.4084541124224481, -1.4856950160464457,
-1.5610214799230007, -1.6364023650343413, -1.7121523543398602,
-1.7857973795759545, -1.8635683428033427, -1.944623534632176, -2.0264219860427,
-2.1049255177639425, -2.1768933759590596, -2.255827277174643,
-2.3302781542474524, -2.4039817330049225, -2.479053504956087,
-2.555332654727632, -2.630548530655701, -2.701593830968021, -2.778090394992306,
-2.849689351723903, -2.9254366487587924, -2.995742320062854,
-3.0686134605708735, -3.1367161877010528, -3.213193391943208,
-3.2821888381603923, -3.3539292526016613, -3.4180623064265405,
-3.4927420018875073, -3.5649783223153935, -3.629213577915999,
-3.7043290731994527, -3.7738187080514987, -3.8415390899251625,
-3.9103960616231217, -3.981495262353599, -4.050231152464092, -4.119848390751093,
-4.191534686702461, -4.25729265794795, -4.323133481741637, -4.389228714109889,
-4.4594408407947705, -4.525585450862313, -4.59595889197424, -4.66348026858393,
-4.727763774509871, -4.791721228824581, -4.854915604163175, -4.921089080518156,
-4.983852939292166, -5.04899681951151, -5.112607215353015, -5.17463432369005,
-5.239460000767522, -5.300430505999815, -5.358008505160151, -5.421302490075876,
-5.481439958046279, -5.540899337266974, -5.599576158720867, -5.664322664355329,
-5.722712940625534, -5.780160636819287, -5.837611083756926, -5.894870868705373,
-5.9536326675524265, -6.0092472713615575, -6.067259966657716,
-6.122043607958796, -6.177782049829729, -6.234713235715359, -6.2881633376372115,
-6.343912468743575, -6.396419637305266, -6.453329672991067, -6.507488523758355,
-6.559371174322699, -6.6084708610826315, -6.662649710205197, -6.71429344395232,
-6.764311671754301, -6.814727236669642, -6.868540801582325, -6.916598444803448,
-6.965928847855524, -7.014471289080296, -7.065486778931989, -7.1139908134050085,
-7.160770219073195, -7.208861712599971, -7.254797981444787, -7.302162264218682,
-7.345519824931329, -7.388829116323121, -7.435582583171289, -7.48416276775139,
-7.527250605202385, -7.570355597605608, -7.612493316252749, -7.654692772946869,
-7.700238554923365, -7.743776705669694, -7.783359884650832, -7.825857050790906,
-7.8706276581483205, -7.911437684207513, -7.950699075154724, -7.992401015051904,
-8.032097928563463, -8.06789479932381, -8.110998720582396, -8.149149639655302,
-8.188509480408996, -8.227876432866466, -8.265911719930614, -8.302888006327915,
-8.343814084135419, -8.382643923757561, -8.420005784590463, -8.453594245371303,
-8.490702826526226, -8.528599366365064, -8.564155151667723, -8.601394564777433,
-8.636901957839921, -8.668844457860136, -8.704322524615755, -8.739863500867639,
-8.776769006202631, -8.809505344613248, -8.842384857153677, -8.875337081786103,
-8.909965228173492, -8.942909287170883, -8.9756687913089, -9.008412038710361,
-9.039467708365244, -9.070494979358497, -9.101473923379185, -9.13107168644742,
-9.160741255001703, -9.190800368496582, -9.22135865561495, -9.249320347435518,
-9.278229520261542, -9.310080972876756, -9.33442891364968, -9.363684904975658,
-9.39076183164832, -9.420550687356181, -9.448597481995533, -9.476663221848595,
-9.503912351627658, -9.531126817221443, -9.55871197094909, -9.584323258880406,
-9.610500344533811, -9.635287409093328, -9.6581664138447, -9.6815725609792,
-9.705253344029648, -9.728772091367533, -9.753672924697586, -9.777244776976262,
-9.795675264887812, -9.82070527419502, -9.843302946145027, -9.865513184071686,
-9.888180723033837, -9.905329826404225, -9.928049563271923, -9.950788536162658,
-9.968356915414862, -9.991667148570729, -10.007694410865753, -10.02434344395731,
-10.040329896774255, -10.061930978836765, -10.07867180076142,
-10.095421076065838, -10.112169411684311, -10.130611136905152,
-10.146593507726045, -10.164335625575253, -10.180375220015684,
-10.19533279114083, -10.211344376367355, -10.226741248352264,
-10.242178945006474, -10.256966871990064, -10.270602167230358,
-10.286045470069888, -10.296067664134682, -10.311050105316802,
-10.32469267299504, -10.337500958927109, -10.346038668421622,
-10.357425506973593, -10.370314594355667, -10.381741800911579,
-10.391628444609328, -10.403033697930681, -10.408874170410353,
-10.418149103826025, -10.423725764623583, -10.436543308758738,
-10.442156938527617, -10.451926329238313, -10.457537831497104,
-10.466904202415876, -10.472371832104965, -10.476286520531145,
-10.48380277225473, -10.48944547934829, -10.496975773710382, -10.49896964373042,
-10.504512934578784, -10.510258770943683, -10.514024629329443,
-10.517716935580324, -10.519532957492546, -10.523380302564586,
-10.527157115994989, -10.52532916640576, -10.527157115994989,
-10.52909931090868, -10.530984691620892, -10.53287026014497,
-10.531034402940119, -10.53287026014497, -10.53287026014497, -10.53287026014497,
-10.534713224485767, -10.534756003099286, -10.53287026014497,
-10.529205624142506, -10.527270932216865, -10.525449627416553,
-10.52169453207786, -10.52169453207786, -10.517940443660144,
-10.517940443660144, -10.51410620147273, -10.512311429442438,
-10.50856031887744, -10.506685272612621, -10.502936263549664,
-10.499079420599642, -10.49532657586526, -10.491698843204722,
-10.486086179971895, -10.480478013370531, -10.476438884474282,
-10.472849889244188, -10.467074009329632, -10.457724992266025,
-10.45399018236726, -10.452123855919485, -10.44445474324496,
-10.439080872751124, -10.436769976177082, -10.423725764623585,
-10.421865930771684, -10.41045283149019, -10.406741175781676,
-10.40088939542301, -10.393783901184603, -10.385765576376306,
-10.374351166181127, -10.371010855531031, -10.3629414170994,
-10.349704330855761, -10.349704330855761, -10.337910366815693,
-10.330172677690804, -10.326087784967237, -10.311050105316806,
-10.301501363105594, -10.289660602777566, -10.286546944643769,
-10.278321785133361, -10.263430141116103, -10.25641161715702,
-10.246905210516237, -10.233868994055054, -10.2255783029342,
-10.214874563564068, -10.202558240573625, -10.192500155745604,
-10.180701396167303, -10.169940522843197, -10.159143827219234,
-10.146372982038377, -10.136326798672275, -10.121331010117952,
-10.111288095813782, -10.098758525104598, -10.086227380444075,
-10.073698596562307, -10.060425039878133, -10.047035463391472,
-10.035235491426713, -10.021131459838614, -10.007694410865756,
-9.993663133081789, -9.980237723255053, -9.967319578610592, -9.952906187383505,
-9.939957395239349, -9.925044489989517, -9.909061655669927, -9.894556902091944,
-9.878591222109979, -9.864082579770315, -9.848163629494982, -9.833677974708921,
-9.817955209439399, -9.800858854460017, -9.786449139694106, -9.768063653652378,
-9.752390150502533, -9.735266599338365, -9.717076938351447, -9.702644232057684,
-9.68204280761959, -9.667663650234276, -9.647802220597756, -9.627156177062638,
-9.608346695685151, -9.596112995499173, -9.576643863477019, -9.55727240001247,
-9.539937624524205, -9.51850772640966, -9.501187655071583, -9.483020071367172,
-9.464167041330516, -9.444502578268123, -9.426439900778025, -9.403519322696392,
-9.38476490203963, -9.362692457993722, -9.343654560909854, -9.321634207779287,
-9.303136570374653, -9.281202606496805, -9.260380069166873, -9.239551000315023,
-9.217373651555796, -9.193625116475971, -9.175074098959266, -9.153244346068625,
-9.131071686447422, -9.107111391496524, -9.086627307384425, -9.061275702171892,
-9.039275808526083, -9.01749657128425, -8.992041910260573, -8.968417502889308,
-8.948503232840908, -8.924745529906014, -8.895773200183283, -8.87352852590067,
-8.848291833343465, -8.822641245826011, -8.800245128179212, -8.77322867581048,
-8.74950630895809, -8.723199685416786, -8.697388099229533, -8.675722121853632,
-8.649410596905403, -8.620850016699944, -8.59166725148004, -8.567454714995618,
-8.539924634527019, -8.514064643908377, -8.485810090220053, -8.45670374472343,
-8.427615593459548, -8.399095465499558, -8.372021231918989, -8.343096209169007,
-8.31354461273922, -8.285184802549818, -8.258607269201642, -8.226703827043337,
-8.19864813969042, -8.168048038919395, -8.138859652999203, -8.106850948450568,
-8.0787380971475, -8.048163502062854, -8.017809682284492, -7.984325538898434,
-7.9552931346713605, -7.923074903381121, -7.89237195174655, -7.860064921566636,
-7.827798295006836, -7.794632327970301, -7.761492238550607, -7.730717938954298,
-7.697572858440747, -7.664288411824797, -7.630937769627149, -7.597594980115436,
-7.5646015413160175, -7.531852765405959, -7.499243155938716,
-7.4628676544834045, -7.428458543001446, -7.39768703055596, -7.363217929556452,
-7.3269446763675, -7.292344468975453, -7.257217423847516, -7.221969208104735,
-7.18668701719389, -7.150983661261604, -7.115422110137542, -7.080840406837606,
-7.042739484965488, -7.0092341645728515, -6.969336633484454,
-6.9358283019651115, -6.898925020580686, -6.860681267425338,
-6.8227325748243235, -6.784721603696065, -6.7484948561890405,
-6.709560874781429, -6.67441583664921, -6.635584128554186, -6.595397677505247,
-6.558817405056116, -6.519745606596052, -6.482991123186787, -6.443216203693101,
-6.403408971918221, -6.361408555145341, -6.321841945395439, -6.283080291139624,
-6.241919825058135, -6.20291214970424, -6.160333824252432, -6.122288063564868,
-6.080118176689244, -6.040226314603456, -6.002566233958158, -5.958113862651262,
-5.917235002989392, -5.874625627801624, -5.8320424274282265, -5.789196753196061,
-5.746626809615487, -5.703892744388478, -5.661826156052108, -5.616676127621605,
-5.573858690181673, -5.533817097378911, -5.487974636382266, -5.441335072541193,
-5.399703620484814, -5.35384106608624, -5.30969367649537, -5.2662897194593015,
-5.222372683439071, -5.171717209497482, -5.127053708125021, -5.08205979430638,
-5.037153207643724, -4.989522795345411, -4.944275051499966, -4.897502616239072,
-4.851837869207443, -4.807258538377478, -4.757116546093491, -4.710270972216836,
-4.66410247784173, -4.615471495011052, -4.568171841218687, -4.519640309830263,
-4.474176475361195, -4.4240207941505005, -4.376842690363722, -4.328462159304697,
-4.281656989389846, -4.229378580216229, -4.182704283770019, -4.134153871043464,
-4.0830238779805885, -4.035550132046515, -3.987211021208055, -3.938937149166608,
-3.886123738333277, -3.838815883613222, -3.7865388382506557,
-3.7401379583409953, -3.6903103800390826, -3.6366046908211507,
-3.590817092090267, -3.5385074392121196, -3.4861992140557994,
-3.436638240406814, -3.3873322706261626, -3.3347769004446453,
-3.284264582958759, -3.23122183894533, -3.180195153251606, -3.1292797411018127,
-3.0761033923928944, -3.024585531198385, -2.974456355514891, -2.923263688216336,
-2.869757217689134, -2.8162758482347217, -2.76515162686001, -2.7124861718811593,
-2.663578020518485, -2.609146763916816, -2.558189309702813, -2.5006803222852323,
-2.452274164652299, -2.3965345014892194, -2.3467586356755614,
-2.2930687497033304, -2.2413786053839164, -2.1849364373386817,
-2.132399276553598, -2.080854029425326, -2.023890452560233, -1.9676545609271516,
-1.912924832793425, -1.8592580246695356, -1.8056683892108814,
-1.746151045096346, -1.6921686515313048, -1.6365640412923974,
-1.5801356591449798, -1.5251858189996386, -1.470310217501044,
-1.4121538858324933, -1.3548112610093177, -1.2995458322019389,
-1.2448690730000924, -1.1869100958961158, -1.1271907824138767,
-1.0691800400862885, -1.0143403263114201, -0.9501656304721633,
-0.8946729645615215, -0.8373261556642395, -0.7798049261212452,
-0.7210382284534893, -0.6600106598384883, -0.6059884125540596,
-0.543599396043404, -0.4850808603649941, -0.42498246019309405,
-0.36407714217317877, -0.3057904587298159, -0.2434251801333187,
-0.18292244945581926, -0.12333808814275038, -0.06337130905310673,
0.0016645324100794667, 0.05870095937915032, 0.12517622107742188,
0.1860304307336591, 0.2484019106271309, 0.3085577171469063, 0.37518433229782877,
0.4344838019773203, 0.5011769533163446, 0.5646055516009412, 0.6264888483023805,
0.6921884059336803, 0.7552363960162336, 0.8211033079335868, 0.8869039868178077,
0.9534061129357532, 1.0173751839885379, 1.0820546076606468, 1.1534951005748935,
1.2143320420134067, 1.2831110262797834, 1.353509341532511, 1.4147632658015494,
1.4833283924277634, 1.550341849949395, 1.6173522181343423, 1.6872264152053411,
1.7535583771116894, 1.8207554642972124, 1.8873991028381953, 1.9584581928795783,
2.025797812894845, 2.097094058418109, 2.1664523783427896, 2.2349361872906237,
2.3036434534855776, 2.3742806479729115, 2.4400818143137277, 2.508323691282113,
2.5805370674194656, 2.650262424709243, 2.718604242396319, 2.7891532483486547,
2.8595568158612537, 2.9300199704325105, 2.9973828788248777, 3.068581559837198,
3.1390536917830048, 3.2030058549545597, 3.2781929169003985, 3.3484742741670144,
3.4149243791651207, 3.488949178617877, 3.5579981542684127, 3.6269252537145467,
3.6976217639610196, 3.76901011567716, 3.837464996631947, 3.910070651226133,
3.9820590710499877, 4.050667155377583, 4.123891916663147, 4.1933086575054155,
4.265756035963238, 4.337188943834065, 4.408401507132912, 4.479756831358671,
4.5516112826988095, 4.62520274937646, 4.696568017258648, 4.769061242406737,
4.841346887160121, 4.910826514321321, 4.987789206595259, 5.058605530665938,
5.134354828187375, 5.20296285944433, 5.277161750471748, 5.354343551538408,
5.425881987124881, 5.498720471816426, 5.57358919451121, 5.645991375929993,
5.722893187947384, 5.798459877172157, 5.876452662576753, 5.950234004140038,
6.024279157675622, 6.099192978451255, 6.176216116029698, 6.255247176329078,
6.333667916038948, 6.409550778774854, 6.488051181395564, 6.568716621591408,
6.643312555787485, 6.723217799721644, 6.803120006439351, 6.882526918477365,
6.963638259775109, 7.042525836997784, 7.121606607020647, 7.198368357639598,
7.282936141913606, 7.363370053910209, 7.447388808015724, 7.527924975497326,
7.607485951373194, 7.690646982220674, 7.773016162535047, 7.853719413107705,
7.9406586003662625, 8.021618408488633, 8.103590017369822, 8.188952166589054,
8.27343787360722, 8.355985251546223, 8.43670482785958, 8.526252078087193,
8.609190316108837, 8.694118650754174, 8.775773955169013, 8.863164972563421,
8.948649668495023, 9.033087166409176, 9.116807292618528, 9.201753700940998,
9.283646987746684, 9.368892748164308, 9.453489638945697, 9.540629010448935,
9.622391027851867, 9.709479055363643, 9.792168001388898, 9.87380808743822,
9.96065026367567, 10.045872660823122, 10.130630882165425, 10.211422529018632,
10.293606923880759, 10.37788439636671, 10.461377497054999, 10.548057518022002,
10.6344159718033, 10.723185026094486, 10.810373276723555, 10.893027832448217,
10.974918831954882, 11.062584727471469]
```

[ ]:
```
time_1 = np.arange(len(theta))/frame_rate
popt_1, _ = sp.optimize.curve_fit(quad, time_1, theta)
popt_1
```

[ ]:
```
array([ 0.38865188, -3.67438253, -1.49138986])
```

[ ]:
```
plt.plot(time_1, theta)
plt.plot(time_1, quad(time_1, *popt_1))
plt.xlabel(r"Temps $[s]$")
plt.ylabel(r"$\theta$ $[rad]$")
plt.savefig("m40_t.png")
```

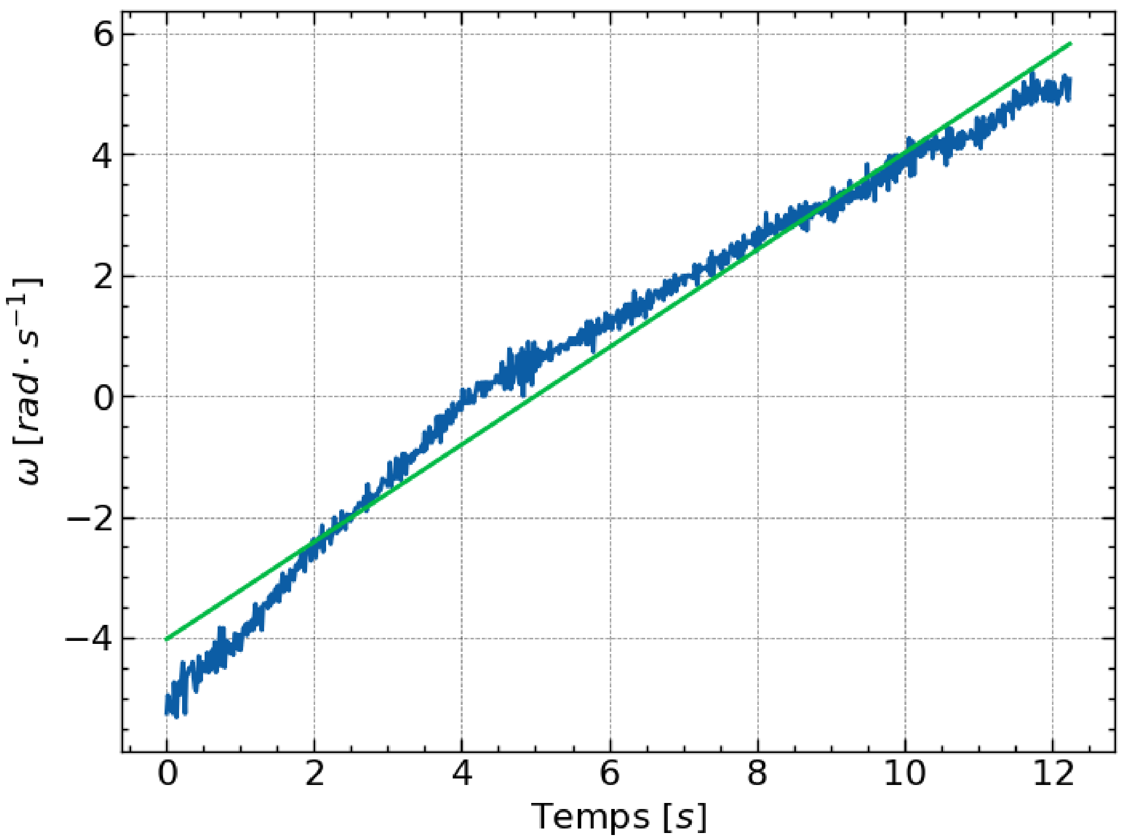


```
[ ]: omega = []

     for i in range(len(theta)-1):
         omega.append((theta[i+1]-theta[i])/(1/frame_rate))
```

```
[ ]: time_2 = np.arange(len(omega))/frame_rate
     popt_2, _ = sp.optimize.curve_fit(lin, time_2, omega)
     popt_2
```

```
[ ]: array([ 0.80432804, -4.02146438])
```

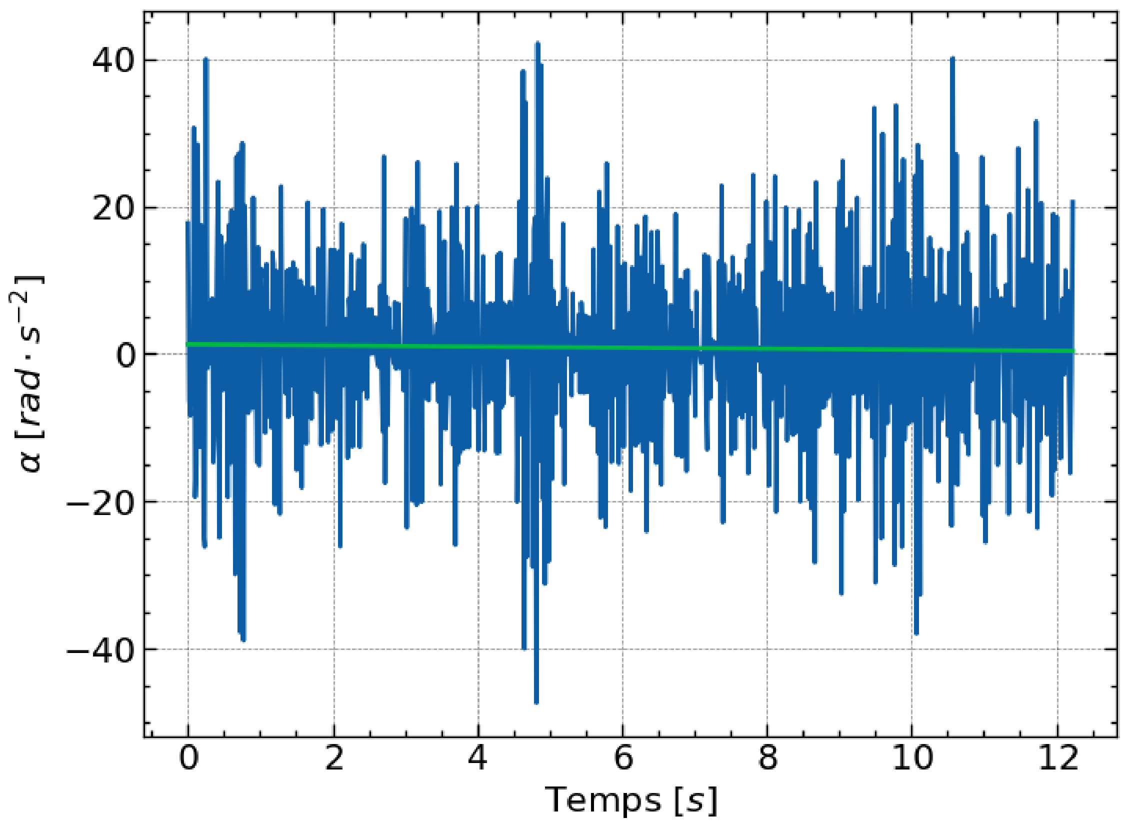
```
[ ]: plt.plot(time_2, omega)
     plt.plot(time_2, lin(time_2, *popt_2))
     plt.xlabel(r"Temps $[s]$")
     plt.ylabel(r"$\omega$ $[rad \cdot s^{-1}]$")
     plt.savefig("m40_o.png")
```



[ ]:

```
alpha = []

for i in range(len(omega)-1):
    alpha.append((omega[i+1]-omega[i])/(1/frame_rate))

time_3 = np.arange(len(alpha))/frame_rate
popt_3, _ = sp.optimize.curve_fit(lin, time_3, alpha)
print(popt_3)

plt.plot(np.arange(len(alpha))/frame_rate, alpha)
plt.plot(np.arange(len(alpha))/frame_rate, lin(time_3, *popt_3))
plt.xlabel(r"Temps $[s]$")
plt.ylabel(r"$\alpha$ $[rad \cdot s^{-2}]$")
plt.savefig("m40_a.png")
```

```
[-0.07261683  1.30070567]
```



```
[ ]: acc_quad = popt_1[0]*2
     acc_lin = popt_2[0]
     print(acc_quad, acc_lin)
     acc = (acc_quad + acc_lin)/2
     Mom = 9.81 * m40 * r_m
     I_m40 = Mom/acc
     print(I_m40)
```

```
0.7773037662037253 0.8043280421534522
0.014885891821090 07
```

1.2 Vidéo m60

```
[ ]: video_path = "m60.mp4"
     cap = cv2.VideoCapture(video_path)
```

```
[ ]: frame_rate = cap.get(cv2.CAP_PROP_FPS)
     print("Frame rate: ", frame_rate)
```

```
Frame rate:  59.78680582027616
```

[ ]:

```
frames = []
while True:
    ret, frame = cap.read()
    if not ret:
        break
    frames.append(frame)
```

[ ]:

```
frame_rate = cap.get(cv2.CAP_PROP_FPS)
print("Frame rate: ", frame_rate)
```

```
Frame rate:  59.78680582027616
```

[ ]:

```
def disp_f(num):
    plt.imshow(frames[num])

widgets.interact(disp_f, num=widgets.IntSlider(min=0, max=len(frames)-1,␣
 ↪step=1))
```

```
interactive(children=(IntSlider(value=0, description='num', max=1008),␣
 ↪Output()), _dom_classes=('widget-intera…
```

[ ]:

```
<function __main__.disp_f(num)>
```

[ ]:

```
analyse = frames[480:886]
```

[ ]:

```
def pos_track(x,y):
    image = analyse[0].copy()
    w, h = 50, 50
    cv2.rectangle(image, (x, y), (x + w, y + h), (255, 0, 0), 4)
    plt.imshow(image)

widgets.interact(pos_track, x=widgets.IntSlider(min=0, max=len(analyse[0]),␣
 ↪step=1), y=widgets.IntSlider(min=0, max=len(analyse[0][0]), step=1))
```

```
interactive(children=(IntSlider(value=0, description='x', max=1080),␣
 ↪IntSlider(value=0, description='y', max=1…
```

[ ]:

```
<function __main__.pos_track(x, y)>
```

[ ]:

```
tracker = cv2.TrackerCSRT_create()

track_frames = analyse.copy()

init_trac_f = track_frames[0]
roi = (331, 573, 50, 50)

tracker.init(init_trac_f, roi)
```

```
results = []
frame_tracked = []
for i in range(1, len(track_frames)):
    # Read the next frame
    frame = track_frames[i]

    # Update the tracker
    success, roi = tracker.update(frame)

    if success:
        # Draw a bounding box around the tracked object
        (x, y, w, h) = tuple(map(int, roi))
        results += [(x,y,w,h)]
        cv2.rectangle(frame, (x, y), (x + w, y + h), (255, 0, 0), 4)
        frame_tracked.append(frame)
```

[ ]:

```
def disp_t(num):
    plt.imshow(frame_tracked[num])

widgets.interact(disp_t, num=widgets.IntSlider(min=0, max=len(frame_tracked)-1,␣
 ↪step=1))
```

```
interactive(children=(IntSlider(value=0, description='num', max=404), Output()),␣
 ↪_dom_classes=('widget-interac…
```

[ ]: `<function __main__.disp_t(num)>`

[ ]:

```
good_results = results
```

[ ]:

```
center_of_rot = (705, 535, 50, 50)
```

[ ]:

```
def pos_from_case(case):
    x,y,w,h = case
    return (x + (x+w))/2, (y + (y+h))/2
```

[ ]:

```
pos_c = pos_from_case(center_of_rot)
pos_r = [pos_from_case(i) for i in good_results]
```

[ ]:

```
theta = []
cur_theta = 0

prev_dir = [pos_r[0][0]-pos_c[0], pos_r[0][1]-pos_c[1]]
for i in range(len(pos_r)):
    next_dir = [pos_r[i][0]-pos_c[0], pos_r[i][1]-pos_c[1]]
    cur_theta += (np.arctan2(next_dir[1], next_dir[0]) - np.
 ↪arctan2(prev_dir[1], prev_dir[0]))
```

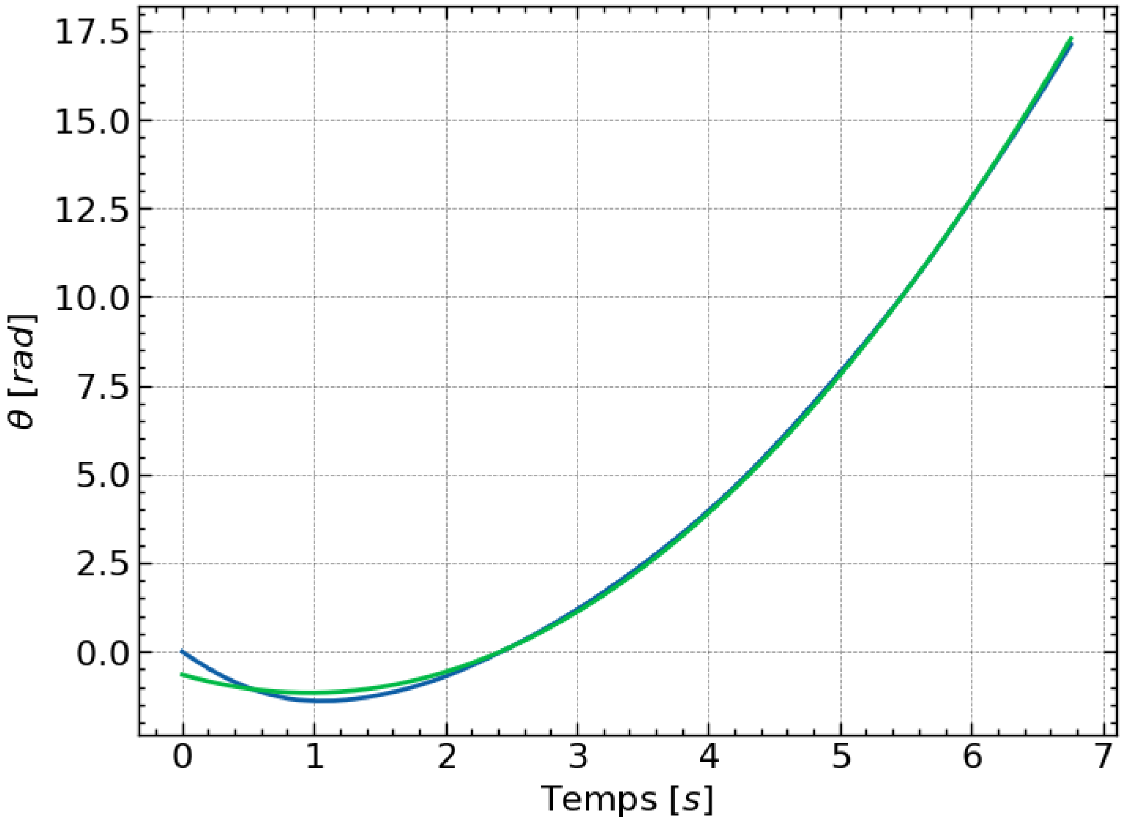
```
    if np.abs(np.arctan2(next_dir[1], next_dir[0]) - np.arctan2(prev_dir[1],␣
 ↪prev_dir[0])) > np.pi:
        cur_theta -= 2*np.pi*np.sign(np.arctan2(next_dir[1], next_dir[0]) - np.
 ↪arctan2(prev_dir[1], prev_dir[0]))
    theta.append(cur_theta)
    prev_dir = next_dir

print(theta)
```

```
[0.0, -0.046383276811095, -0.08874740571573225, -0.13015454387925285,
-0.1689842042016183, -0.20942116835834357, -0.2499888461535571,
-0.28640655690088135, -0.325777893533997, -0.365228466695386,
-0.40142213321366826, -0.4361257163092036, -0.4816343216554446,
-0.5141850004808606, -0.5466382305508484, -0.5805656365961904,
-0.6163902699738131, -0.6521654293676904, -0.6861185272186199,
-0.7160641998941495, -0.7480193006285791, -0.7782772257763795,
-0.8091358123359877, -0.8360815550336964, -0.8640639541386701,
-0.8895513269542765, -0.9169169629105376, -0.9437315393910448,
-0.9704171568935482, -0.9958322101501587, -1.0177042457310117,
-1.0419971862305242, -1.0662848688234565, -1.0862824828005717,
-1.1038887648484175, -1.1214713557866443, -1.141499752778254,
-1.1632870908165416, -1.1825279865082392, -1.2001600026499173,
-1.2157661571794156, -1.2302345170409839, -1.2493546203854056,
-1.2627825006997933, -1.2757993400617889, -1.2895988223581993,
-1.2988987556016642, -1.3134218917500742, -1.3265123279730855,
-1.335818340456728, -1.3465351601697388, -1.3531895666937555,
-1.3622800366919579, -1.3663030592068641, -1.3729406726307534,
-1.3795656959793514, -1.3820670122482275, -1.3860672677464365,
-1.3900583528789132, -1.391322225459495, -1.3939504857202305,
-1.3979334824240626, -1.3979334824240626, -1.4004867478648992,
-1.3992083881883592, -1.4005594071963055, -1.3992083881883592,
-1.3979334824240626, -1.397853915269552, -1.3939504857202305,
-1.391322225459495, -1.3848108308911922, -1.3808146344217602,
-1.3741816798718147, -1.3701652475233546, -1.3649075775977884,
-1.352992270690002, -1.3503656832519846, -1.342492512646501,
-1.3370101190485113, -1.3291347603080956, -1.3215519344380868,
-1.3104995979054384, -1.298558838998383, -1.2907215296913839,
-1.2814071413207833, -1.272093593276141, -1.258666269253 4227,
-1.2478223660418508, -1.2369632155913126, -1.22453135576965 3,
-1.2120627028915405, -1.1991727672655073, -1.1898846130540992,
-1.1751383462831946, -1.1632870908165416, -1.148213733999202,
-1.1298434097472456, -1.1172844627914105, -1.0988532571959444,
-1.08458849842266, -1.069377004917755, -1.0535496488220186, -1.0358075309728112,
-1.020460648105084, -1.0008585147561584, -0.9829833516519808,
-0.9639058891926795, -0.9460826598623715, -0.9275240867835275,
-0.9083390900218236, -0.8886435064698919, -0.8667266969043825,
-0.8479140607262052, -0.8286768359975287, -0.8094480949700187,
-0.7849137317640533, -0.7603002941231836, -0.7371649293041043,
-0.7163403842933307, -0.695630221969278, -0.6692083640595974,
-0.646523078784814, -0.6219977741284732, -0.5955256684229946,
-0.5727881223805067, -0.5460934655124308, -0.5214432882857731,
-0.49684713815146475, -0.4701544935712785, -0.44237998301360903,
-0.41524033568484686, -0.38764694237040187, -0.3610732422420093,
-0.33117215519807663, -0.3036665629171562, -0.2766635208940982,
-0.24606445496169727, -0.21549564856114012, -0.19093566989322364,
-0.16222586794315141, -0.1266521685860118, -0.09876669650781977,
-0.0657412296398876, -0.03803828054443281, -0.006043315407612937,
0.0222658308060101 9, 0.05319591012560520 6, 0.0851485180754831 8,
0.11195068799113317, 0.14930634285161126, 0.18252112524114494,
0.2145246211629032 7, 0.24244949180350384, 0.2755504465859415,
0.306273271851535 1, 0.3400442114564490 3, 0.372827880500997 9,
0.40727052055274404, 0.4426088583338461, 0.474407212138805, 0.51039185336478,
0.5447626672583312, 0.5784763232629788, 0.6127935111767342, 0.6482090900494808,
0.6817137508994562, 0.7196867822763182, 0.7558560004810779, 0.7900919213188424,
0.8257927573777679, 0.8595086956380928, 0.9002761784105959, 0.9373058096661397,
0.9743100774283735, 1.0111880468924581, 1.048281896948854, 1.0910729971219904,
1.1286996923239032, 1.1658827424614677, 1.2040851713308554, 1.2445860674163813,
1.2827524670149573, 1.3255064676953343, 1.3614748682462916, 1.4047762615129173,
1.4431196320227986, 1.4881944163176768, 1.5271488618125402, 1.5690034000268211,
1.6104515299772757, 1.6491009341404848, 1.6942411934745936, 1.7378303875741836,
1.7813762355569567, 1.8226857699277088, 1.863663114465292, 1.907830010467385,
1.9525249207485262, 1.9940233363547435, 2.036809736400123, 2.082817013478457,
2.1263660334200827, 2.1742435922615098, 2.2175724591017865, 2.263018538575726,
2.3067460134689393, 2.3546600866048086, 2.39828258801018, 2.4449385782198503,
2.4916490213086218, 2.536594981083793, 2.585114953800091, 2.630091511700949,
2.6758046188058526, 2.7251039215700477, 2.772956664212374, 2.8234265498073934,
2.872700680717971, 2.9190747679522517, 2.9665067934655616, 3.0170987843306665,
3.068458155901397, 3.1201336434373275, 3.1682470560444242, 3.2180117800801296,
3.2702650869453764, 3.319849657302328, 3.3732908878308594, 3.424698475986398,
3.4751434498180167, 3.527060144467819, 3.5826466651537316, 3.6358457798372785,
3.687789457758615, 3.742754903401012, 3.794327104660705, 3.850577672612401,
3.9053868482870326, 3.9567756311417948, 4.015048953413333, 4.0696394685938095,
4.126251671870523, 4.181102166838778, 4.240498934541614, 4.296438316647696,
4.355484256046659, 4.409949376694892, 4.469945238272568, 4.525056810756751,
4.586734069876147, 4.648951084423588, 4.705760637568422, 4.7700527843627185,
4.828796937377653, 4.889145134742199, 4.949740303349747, 5.013692894177661,
5.075708863518896, 5.135145607289313, 5.20099047146557, 5.262365043244342,
5.328163630121926, 5.390457173927883, 5.457393316812645, 5.521679062860879,
5.587654674758765, 5.651713512727101, 5.719515818739671, 5.785791202792126,
5.84705959087038, 5.915625044517876, 5.984219710330608, 6.051766323798983,
6.115877297295773, 6.186033242910748, 6.251470974362986, 6.319003625024084,
6.388517955752497, 6.457823042825466, 6.527090751816122, 6.596659431115056,
6.665348195428527, 6.738364321826692, 6.806830051824107, 6.876495284866784,
6.948584251397544, 7.018219295411331, 7.090329357880908, 7.157579958054525,
7.231562038088983, 7.3018920093843445, 7.372437841566697, 7.443694675390931,
7.5114868596370075, 7.586968518437194, 7.661733405587553, 7.728793092830241,
7.808348397449227, 7.8765876744602465, 7.950271771454355, 8.022331565916081,
8.099360152999719, 8.172934046475874, 8.243783533661652, 8.32080801967736 5,
8.39348204410687, 8.468824695014467, 8.541857441118324, 8.617766034256496,
8.693968490921174, 8.764109220218838, 8.844895137121643, 8.917071079325112,
8.99487221802285, 9.075537926170352, 9.149587342461937, 9.228837232686425,
9.300952489055897, 9.380447342457034, 9.459566586703861, 9.536979171806774,
9.615735054017021, 9.694936170591184, 9.772303329413054, 9.851322575321953,
9.924703408645751, 10.01283223142665, 10.092921847656445, 10.17426808186768,
10.256409319830258, 10.338306739603212, 10.418304800696859, 10.502705178668592,
10.583601109366903, 10.667056674857733, 10.748321512065052, 10.83293180360898,
10.919413009229036, 11.003342017592358, 11.092503187774556, 11.176212516689674,
11.260784359903049, 11.344906157655092, 11.438274061982325, 11.526819357974883,
11.60956725823391, 11.700061223613062, 11.789451045788514, 11.880252087140553,
11.967071997959826, 12.061782849037947, 12.150278407806423, 12.242386591204362,
12.335707009343668, 12.42025166976902, 12.513723440562835, 12.605753985849919,
12.696971656111556, 12.790498449523106, 12.884463307982179, 12.973641134911919,
13.07199224612972, 13.164933537164005, 13.252614799114376, 13.346998474532919,
13.442323078494303, 13.533286731541114, 13.62562945929491, 13.717485467035791,
13.812100603516797, 13.906302108653607, 14.001338660224324, 14.094546775470327,
14.188222652413355, 14.281102593337057, 14.373252607744359, 14.470717737382294,
14.565291743300232, 14.655372681979635, 14.757707252588947, 14.853367316513143,
14.947999903330103, 15.042003639179173, 15.1369039716264, 15.234185944352106,
15.330995659598475, 15.43046426884486, 15.528062643284018, 15.624382980254541,
15.722004312433981, 15.816404735784195, 15.917854733254305, 16.017638333131718,
16.11753450950896, 16.212663222255962, 16.31997375207554, 16.417341137079045,
16.52134165184437, 16.62510142477032, 16.721722780962967, 16.821920146436412,
16.92764228525224, 17.029812642037047, 17.12845631219424]
```

[ ]:
```
time_1 = np.arange(len(theta))/frame_rate
popt_1, _ = sp.optimize.curve_fit(quad, time_1, theta)
popt_1
```

[ ]:
```
array([ 0.55067263, -1.06492991, -0.65186379])
```

[ ]:
```
plt.plot(time_1, theta)
plt.plot(time_1, quad(time_1, *popt_1))
plt.xlabel(r"Temps $[s]$")
plt.ylabel(r"$\theta$ $[rad]$")
plt.savefig("m60_t.png")
```



[ ]:

```
omega = []

for i in range(len(theta)-1):
    omega.append((theta[i+1]-theta[i])/(1/frame_rate))
```

[ ]:

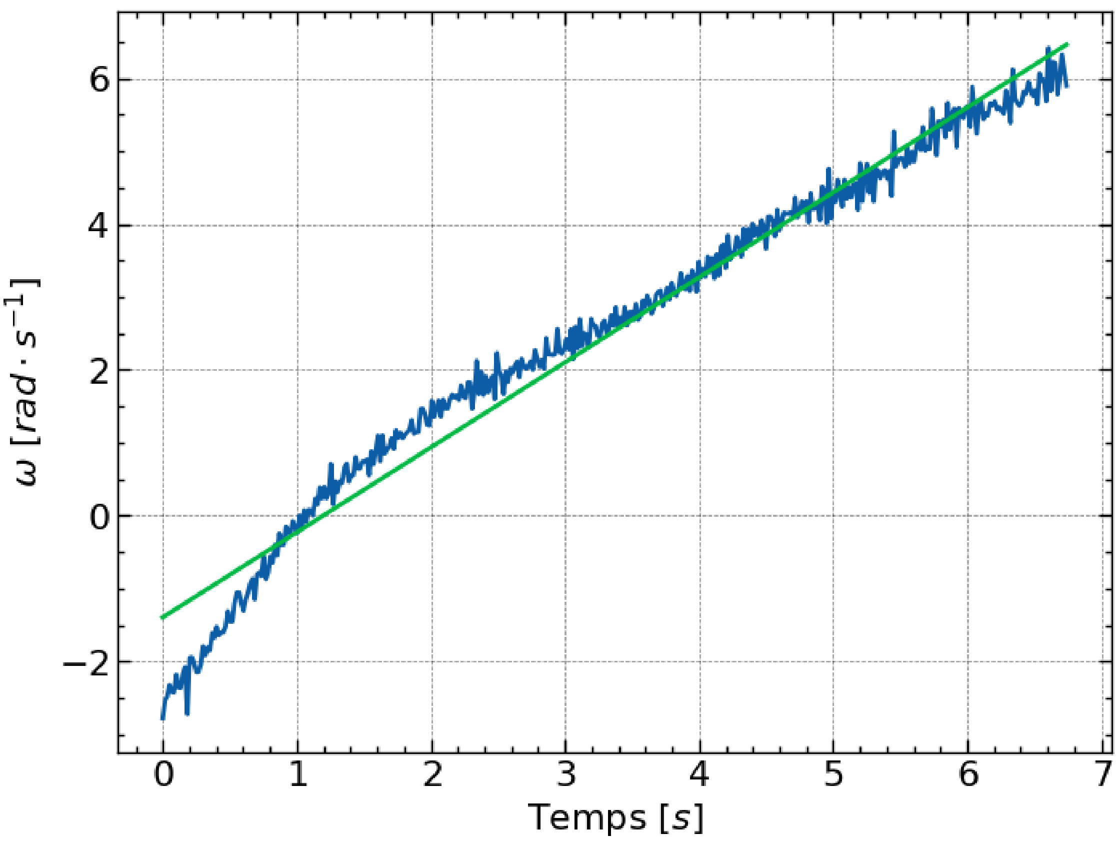
```
time_2 = np.arange(len(omega))/frame_rate
popt_2, _ = sp.optimize.curve_fit(lin, time_2, omega)
popt_2
```

[ ]:

```
array([ 1.16505327, -1.39179803])
```

[ ]:

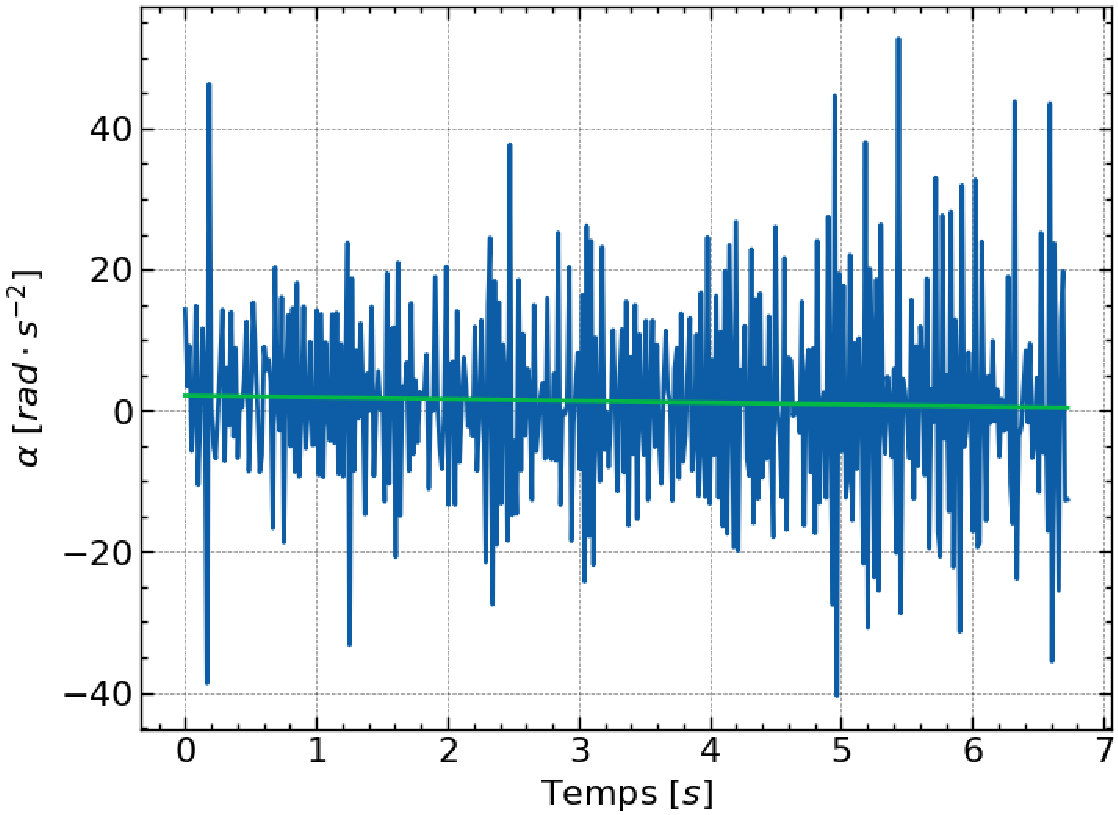
```
plt.plot(time_2, omega)
plt.plot(time_2, lin(time_2, *popt_2))
plt.xlabel(r"Temps $[s]$")
plt.ylabel(r"$\omega$ $[rad \cdot s^{-1}]$")
plt.savefig("m60_o.png")
```



[ ]:

```
alpha = []

for i in range(len(omega)-1):
    alpha.append((omega[i+1]-omega[i])/(1/frame_rate))

time_3 = np.arange(len(alpha))/frame_rate
popt_3, _ = sp.optimize.curve_fit(lin, time_3, alpha)
print(popt_3)

plt.plot(np.arange(len(alpha))/frame_rate, alpha)
plt.plot(np.arange(len(alpha))/frame_rate, lin(time_3, *popt_3))
plt.xlabel(r"Temps $[s]$")
plt.ylabel(r"$\alpha$ $[rad \cdot s^{-2}]$")
plt.savefig("m60_a.png")
```

```
[-0.25749748  2.15202835]
```



```
[ ]: acc_quad = popt_1[0]*2
     acc_lin = popt_2[0]
     print(acc_quad, acc_lin)
     acc = (acc_quad + acc_lin)/2
     Mom = 9.81 * m60 * r_m
     I_m60 = Mom/acc
     print(I_m60)
```

```
1.1013452537778434 1.1650532741707633
0.015582431582306796
```

1.3 Vidéo p70

```
[ ]: video_path = "p70.mp4"
     cap = cv2.VideoCapture(video_path)
```

```
[ ]: frame_rate = cap.get(cv2.CAP_PROP_FPS)
     print("Frame rate: ", frame_rate)
```

```
Frame rate:  30.0
```

[ ]:

```
frames = []
while True:
    ret, frame = cap.read()
    if not ret:
        break
    frames.append(frame)
```

[ ]:

```
frame_rate = 120
print("Frame rate: ", frame_rate)
```

```
Frame rate:  120
```

[ ]:

```
def disp_f(num):
    plt.imshow(frames[num])

widgets.interact(disp_f, num=widgets.IntSlider(min=0, max=len(frames)-1,␣
 ↪step=1))
```

```
interactive(children=(IntSlider(value=0, description='num', max=2197),␣
 ↪Output()), _dom_classes=('widget-intera…
```

[ ]:

```
<function __main__.disp_f(num)>
```

[ ]:

```
analyse = frames[532:1879]
```

[ ]:

```
def pos_track(x,y):
    image = analyse[0].copy()
    w, h = 50, 50
    cv2.rectangle(image, (x, y), (x + w, y + h), (255, 0, 0), 4)
    plt.imshow(image)

widgets.interact(pos_track, x=widgets.IntSlider(min=0, max=len(analyse[0]),␣
 ↪step=1), y=widgets.IntSlider(min=0, max=len(analyse[0][0]), step=1))
```

```
interactive(children=(IntSlider(value=0, description='x', max=1080),␣
 ↪IntSlider(value=0, description='y', max=1…
```

[ ]:

```
<function __main__.pos_track(x, y)>
```

[ ]:

```
tracker = cv2.TrackerCSRT_create()

track_frames = analyse.copy()

init_trac_f = track_frames[0]
roi = (1054, 666, 50, 50)

tracker.init(init_trac_f, roi)
```

```
results = []
frame_tracked = []
for i in range(1, len(track_frames)):
    # Read the next frame
    frame = track_frames[i]

    # Update the tracker
    success, roi = tracker.update(frame)

    if success:
        # Draw a bounding box around the tracked object
        (x, y, w, h) = tuple(map(int, roi))
        results += [(x,y,w,h)]
        cv2.rectangle(frame, (x, y), (x + w, y + h), (255, 0, 0), 4)
        frame_tracked.append(frame)
```

[ ]:

```
def disp_t(num):
    plt.imshow(frame_tracked[num])

widgets.interact(disp_t, num=widgets.IntSlider(min=0, max=len(frame_tracked)-1,␣
 ↪step=1))
```

```
interactive(children=(IntSlider(value=0, description='num', max=1345),␣
 ↪Output()), _dom_classes=('widget-intera…
```

[ ]:

```
<function __main__.disp_t(num)>
```

[ ]:

```
good_results = results
```

[ ]:

```
center_of_rot = (722, 535, 50, 50)
```

[ ]:

```
def pos_from_case(case):
    x,y,w,h = case
    return (x + (x+w))/2, (y + (y+h))/2
```

[ ]:

```
pos_c = pos_from_case(center_of_rot)
pos_r = [pos_from_case(i) for i in good_results]
```

[ ]:

```
theta = []
cur_theta = 0

prev_dir = [pos_r[0][0]-pos_c[0], pos_r[0][1]-pos_c[1]]
for i in range(len(pos_r)):
    next_dir = [pos_r[i][0]-pos_c[0], pos_r[i][1]-pos_c[1]]
    cur_theta += (np.arctan2(next_dir[1], next_dir[0]) - np.
 ↪arctan2(prev_dir[1], prev_dir[0]))
```

```
    if np.abs(np.arctan2(next_dir[1], next_dir[0]) - np.arctan2(prev_dir[1],␣
 ↪prev_dir[0])) > np.pi:
        cur_theta -= 2*np.pi*np.sign(np.arctan2(next_dir[1], next_dir[0]) - np.
 ↪arctan2(prev_dir[1], prev_dir[0]))
    theta.append(cur_theta)
    prev_dir = next_dir

print(theta)
```

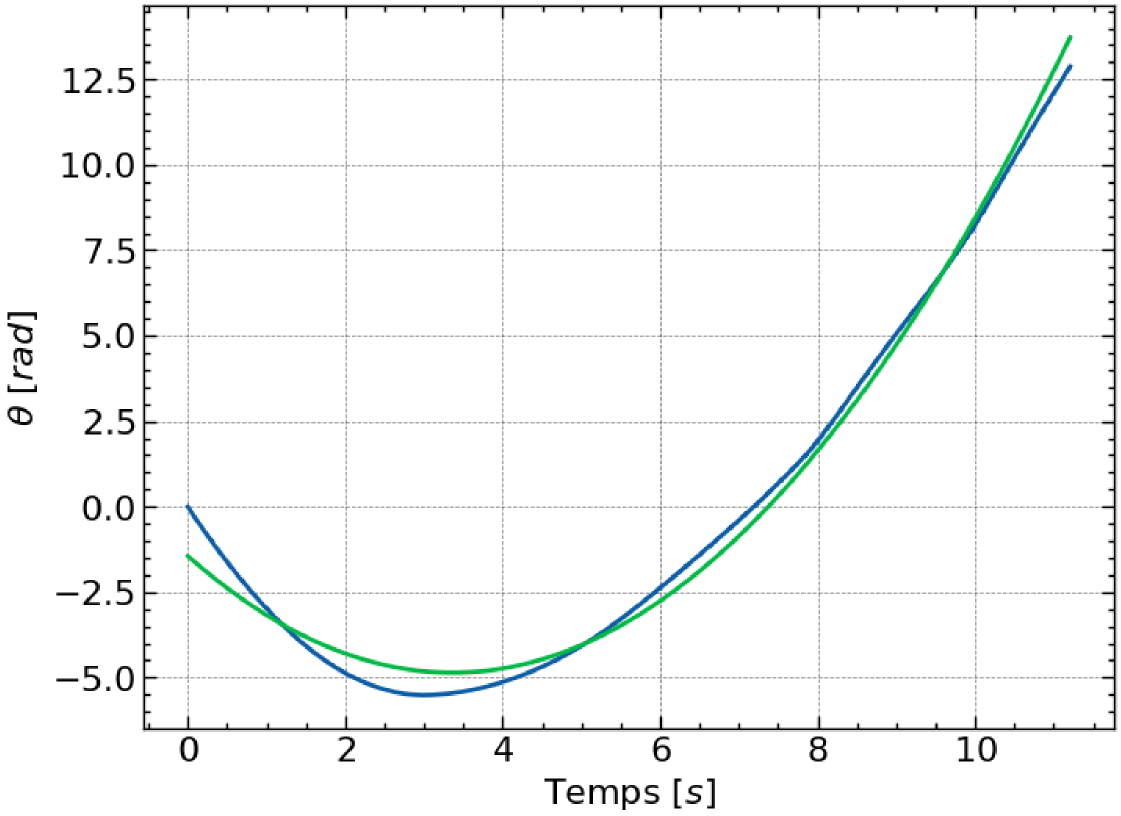
```
[0.0, -0.030996209516437223, -0.041607678625293465, -0.08577825723190313,
-0.11760688824400375, -0.14936612093480417, -0.15979701994223242,
-0.20179540976564356, -0.23684876094692023, -0.26460644543076717,
-0.29132120855139665, -0.32202142029908076, -0.35123193877102793,
-0.36373966349080733, -0.40686576317582307, -0.43203768439805407,
-0.4615489084633472, -0.49228053543770073, -0.5206559015693326,
-0.5468670678674077, -0.5599578604963822, -0.6014281516399573,
-0.6128881486869464, -0.6559984950971667, -0.6823357851508924,
-0.710442936493695, -0.7377797241077169, -0.7695285762939448,
-0.7962414902935852, -0.8068210177068809, -0.849237403983332,
-0.8722322586937201, -0.9008694640870352, -0.9294279381150347,
-0.9526168120626443, -0.9830442420968716, -0.9830442420968716,
-1.0323708685132071, -1.0302119268330836, -1.0856358497400365,
-1.1152188699827785, -1.1429397582678464, -1.1666876103959356,
-1.1905127884702704, -1.2145300675992683, -1.2165079224726258,
-1.2687367995227739, -1.296861614917209, -1.319094587956903,
-1.3456270835645754, -1.3721768076130818, -1.3861579287101 53,
-1.420603514419942, -1.43453824514907, -1.4704623263864116, -1.4982329629798328,
-1.5241620982243986, -1.5417464248481636, -1.5722889696510844,
-1.5964988509392004, -1.5982877434460845, -1.6449788101601472,
-1.647714588533387, -1.6967065977638436, -1.7201784268290374,
-1.7431063149125061, -1.7689082997531254, -1.7947595105359657,
-1.8032740268685077, -1.8392471892226798, -1.8649270204523782,
-1.8878118351294821, -1.9136001574764192, -1.939375915048833,
-1.9637677305394623, -1.9710008402723718, -2.008416217740451,
-2.0318839529259862, -2.0550970296932194, -2.0799483821003903,
-2.104205024633629, -2.1127197084180134, -2.150200473427963,
-2.1705457247005118, -2.1705457247005118, -2.2170906926219702,
-2.2418843009798137, -2.263984825010672, -2.286141581354363, -2.311024397925653,
-2.324027985562373, -2.356610530847874, -2.380104901559868, -2.401071684477334,
-2.412773013961784, -2.4443209484084423, -2.4649370716719594,
-2.4890265155670486, -2.5117887551155054, -2.514215684152144,
-2.5558431982452348, -2.576466335614994, -2.599489154894071,
-2.6206150339440053, -2.6433449116268424, -2.6642163367571214,
-2.672575123610732, -2.706039531958585, -2.729102380057806, -2.7500361687241037,
-2.7689813441586106, -2.792135692454159, -2.813440751109889,
-2.8222889019785113, -2.854184322898236, -2.8777489538201597,
-2.899251015806391, -2.920753077792622, -2.9401880001711973,
-2.9468341190833742, -2.9786999355463846, -2.992093976925419,
-2.992093976925419, -3.040326859895957, -3.062356656019984, -3.0819681323018147,
-3.1047990120318536, -3.119863989250785, -3.132182069079069, -3.160741196188155,
-3.174098841783267, -3.2064715492295814, -3.2217950109251836,
-3.241174039240679, -3.258414420470124, -3.2779609408951815,
-3.2999527284764496, -3.2999527284764496, -3.3397867851571945,
-3.358203823619702, -3.3762612775491765, -3.394672768544607, -3.414213054050839,
-3.4277818490247474, -3.4567752554129125, -3.4629423007197424,
-3.490293123671494, -3.5054886799934235, -3.5267467517691093,
-3.543510785838257, -3.5649594351712603, -3.5831418573730542,
-3.5908568762383144, -3.6166942735775307, -3.634720453550722,
-3.6519032948139944, -3.6691465452788226, -3.687628908168092,
-3.706748328871752, -3.7145088799731396, -3.743649834204551, -3.758037762241813,
-3.7791479229186833, -3.7926751737175204, -3.810929099188304,
-3.830175216104975, -3.831201299633033, -3.8638096784142544,
-3.8638096784142544, -3.8985325443309344, -3.914886895395393,
-3.9325234473839985, -3.9488618022223783, -3.965156881725542,
-3.9713065590413024, -3.9958301007960455, -4.0073422403902015,
-4.033420182787745, -4.043767264691117, -4.063056575748693, -4.07902978989546,
-4.095867523273277, -4.111030427568393, -4.111030427568393, -4.138963725869493,
-4.158920558491163, -4.1678865509018515, -4.187207223965322, -4.201954352967459,
-4.199974872687074, -4.229851384433813, -4.238549869809126, -4.26199120581026,
-4.2769161492263965, -4.289744135149122, -4.302430211615567, -4.319625749047891,
-4.336530639493265, -4.336530639493265, -4.359850145316184, -4.377194310614604,
-4.396216055270028, -4.402537403025008, -4.419652365547026, -4.427713845728228,
-4.4298034122893615, -4.455531564207449, -4.464641989840738, -4.486262137605689,
-4.495000564821353, -4.513534206628153, -4.520191915686048, -4.535568039975923,
-4.543148131937738, -4.565724411007812, -4.574405003174897, -4.589738813328239,
-4.601068905051916, -4.616384139035467, -4.626476894925348, -4.627723333026294,
-4.653548175444618, -4.6523759934217885, -4.676165020144105,
-4.6874660658724645, -4.701538962628358, -4.711822172076628, -4.725021674413815,
-4.73067620828503, -4.7447340558274815, -4.7587885705322215,
-4.7587885705322215, -4.779343948279474, -4.7925779280409735,
-4.802992430652261, -4.8148627536959605, -4.817732418143208, -4.835688610569573,
-4.851390030626406, -4.858917271659463, -4.872792518466661, -4.876224062326329,
-4.8861118483251085, -4.8977603169843285, -4.90573660161372, -4.919070765714766,
-4.932415010812657, -4.940254624278268, -4.951941101666682, -4.956480433247911,
-4.971375160062552, -4.971375160062552, -4.992066577628004, -4.9949912239996355,
-5.0082378134067085, -5.019945232001733, -5.025708285967699, -5.037524969614598,
-5.047801403071057, -5.053657587866037, -5.0551259778736055, -5.074106960186105,
-5.07851492491167, -5.091613979012545, -5.098962775924293, -5.109100441909863,
-5.112012105874148, -5.123649117979892, -5.132365388585489, -5.132365388585489,
-5.150029790344368, -5.155833723389221, -5.164527127359813, -5.170313849744863,
-5.178979679167828, -5.178979679167828, -5.196656185812083, -5.2024130869991465,
-5.211030290751824, -5.215103562296044, -5.228195013107223, -5.2286865188396865,
-5.240105800198956, -5.246339326292318, -5.246339326292318, -5.2577169387760145,
-5.2662160645567955, -5.271371349413646, -5.281590769240462, -5.287234906810919,
-5.285498848282357, -5.299534563396807, -5.303028609275718, -5.311414156954993,
-5.316697579146086, -5.320238009545481, -5.325823678122241, -5.334170096023319,
-5.33694354687783, -5.343287055798621, -5.348816800071094, -5.354328304034775,
-5.359730203163625, -5.365229582826728, -5.371596491355166, -5.369782620747215,
-5.382525289442429, -5.384345923786557, -5.3898275118010694, -5.39526787478293,
-5.399816025597742, -5.40252386138714, -5.411607951340661, -5.413454759052666,
-5.420686793155644, -5.42254060787161, -5.426046994434337, -5.428941234961519,
-5.43429195678623, -5.438822776758779, -5.436155687702182, -5.4425595044388615,
-5.443351171970956, -5.449749885586281, -5.455044190726521, -5.45691997609047,
-5.458798629962942, -5.464451735061293, -5.465202467090367, -5.467834429560925,
-5.4742364328430435, -5.47685584007078, -5.4794691530366615, -5.483254965128475,
-5.482539675311388, -5.483254965128475, -5.487757998251401, -5.487051603855774,
-5.492256640426505, -5.494849787708221, -5.494158074867137, -5.494158074867137,
-5.4979688473648105, -5.498654259469307, -5.503145621108814, -5.503145621108814,
-5.502469156049955, -5.505053765679312, -5.505724472847735, -5.506964476468308,
-5.507631986637769, -5.510203808434632, -5.512113184097685, -5.510203808434632,
-5.510203808434632, -5.512113184097685, -5.511454635535957, -5.5140250714341565,
-5.5140250714341565, -5.512113184097684, -5.512113184097684, -5.510793561407171,
-5.511454635535956, -5.513369752797315, -5.513369752797315, -5.50887774467676,
-5.509542043825351, -5.511454635535956, -5.510793561407171, -5.510793561407171,
-5.50887774467676, -5.509542043825351, -5.504380492275997, -5.502469156049954,
-5.505053765679311, -5.505053765679311, -5.504380492275997, -5.501240051457484,
-5.5005604089067734, -5.498654259469306, -5.49796884736481, -5.496062146627984,
-5.496750716261438, -5.494158074867135, -5.493463724689272, -5.490357851550943,
-5.488953963454374, -5.487757998251397, -5.487757998251397, -5.487051603855771,
-5.485151935088281, -5.483254965128472, -5.480642331814369, -5.478747717504862,
-5.476128609305223, -5.476128609305223, -5.4697212895807485,
-5.4723470231913485, -5.467089521618712, -5.467089521618712, -5.465950373224284,
-5.4606801456376886, -5.458036111408406, -5.454275785093751, -5.454275785093751,
-5.451625731124916, -5.448969832007458, -5.447876958803494, -5.442559504438857,
-5.43989214934349, -5.436155687702177, -5.436155687702177, -5.4334829355484295,
-5.429758896267437, -5.429758896267437, -5.423369641177103, -5.419851930812614,
-5.417998472176258, -5.414300883804822, -5.411607951340657, -5.408909655381865,
-5.408909655381865, -5.399816025597737, -5.399816025597737, -5.397102963131241,
-5.3925502650216375, -5.387997038295406, -5.387997038295406, -5.378889753984916,
-5.378889753984916, -5.376154208592235, -5.369782620747211, -5.367039114938149,
-5.364290795384849, -5.357926141404942, -5.352410092070631, -5.351574847425928,
-5.347847908861014, -5.343287055798617, -5.340515421166996, -5.333173752393394,
-5.328610128808762, -5.327840014401263, -5.32227719421976, -5.31772314284676,
-5.319489469130276, -5.310378844423402, -5.306537366806254, -5.300225533225484,
-5.2946077556928675, -5.291793127017753, -5.286152574428794, -5.285498848282353,
-5.272479633385961, -5.270759113150137, -5.267930668809926, -5.265098812041276,
-5.259425018817422, -5.256583161935683, -5.247491128311891, -5.241794476887975,
-5.240105800198954, -5.233226940672014, -5.228686518839686, -5.22129022893369,
-5.217222122883436, -5.2122414632311695, -5.209376815539885, -5.199535816565664,
-5.193774239648236, -5.189256618545465, -5.180599888504407, -5.174818381047043,
-5.177710139508118, -5.167421407825556, -5.163232181770837, -5.152932558316357,
-5.145800558299147, -5.142889793244196, -5.135493579368644, -5.132582199505961,
-5.12675568925738, -5.123649117979891, -5.110633007759795, -5.110633007759795,
-5.097560851760244, -5.0931232766820775, -5.087281383521073, -5.078574922683108,
-5.078514924911669, -5.063897832702741, -5.060973914424101, -5.060973914424101,
-5.046354572983163, -5.04050788609145, -5.034593838302728, -5.030238670712848,
-5.021355704693106, -5.021355704693106, -5.006700119614887, -5.008237813406706,
-4.990505792052434, -4.984652620247241, -4.977224935889376, -4.972711988479796,
-4.968189206353905, -4.966858144104656, -4.949016627412405, -4.9399083219355235,
-4.936983211580907, -4.929867742295152, -4.921991023713983, -4.91615289154469,
-4.90573660161372, -4.902369303360398, -4.890228128078296, -4.881496764406863,
-4.8768754431617705, -4.86934478165427, -4.858917271659465, -4.853135730986669,
-4.852533915145219, -4.834573345008222, -4.82814551801695, -4.822390009686458,
-4.812351858685677, -4.800876765573738, -4.800122366516867, -4.785041243062144,
-4.782190298568178, -4.769151417835233, -4.759769259866936, -4.7532150713818595,
-4.743818047064516, -4.734419694717915, -4.727832990097366, -4.724047412946787,
-4.71182217207663, -4.704325344063273, -4.693011948467758, -4.686388003974577,
-4.677820347330003, -4.668415191887126, -4.668415191887126, -4.646494205198044,
-4.64768518870774, -4.627723333026295, -4.624493585931479, -4.6151074606959845,
-4.60643384212729, -4.597072557771028, -4.591721378423779, -4.577681487605243,
-4.568356124500358, -4.5588229992914835, -4.551533175256922, -4.542240860672332,
-4.534621601514629, -4.519174448768674, -4.510926059482553, -4.510425218020462,
-4.493395892812822, -4.482115004792069, -4.468800978477242, -4.46008089519508,
-4.451746557348049, -4.442009484766652, -4.443748605551059, -4.422384722785031,
-4.4136454952345305, -4.400431421444062, -4.3903843575555, -4.383780035449581,
-4.377194310614605, -4.360221683649097, -4.355970225341519, -4.336940748194654,
-4.330403337665155, -4.319726187323668, -4.311176609614514, -4.298342060957528,
-4.287639953671874, -4.283358011787037, -4.266227888247196, -4.2554860869968945,
-4.2424885200262, -4.234044366623254, -4.221195100479379, -4.2106329194078995,
-4.208362817308156, -4.184707172042887, -4.184518615149228, -4.160809447176188,
-4.151508768593482, -4.141094992204218, -4.129839815319265, -4.119839455172579,
-4.110684430765369, -4.095109972156454, -4.0816040005266, -4.0714635735768745,
-4.057491594408001, -4.049079770867811, -4.042301125287225, -4.024008214567086,
-4.017741402278923, -3.9988097465242616, -3.9884262671850252,
-3.976851685698585, -3.964599001727628, -3.957750436318697, -3.949535108986038,
-3.9275858381677753, -3.9166680320453815, -3.9153970284495023,
-3.8893352966168284, -3.876860996606902, -3.864341206367329,
-3.8546478631311363, -3.840092308152904, -3.833154067148114,
-3.8302775597746592, -3.8031676641492784, -3.7905057971706273,
-3.7769451905991156, -3.7672403286353844, -3.7545832542946833,
-3.753774269089343, -3.7285175893145968, -3.7165852734124316,
-3.7009226068939394, -3.6852519518855056, -3.674396221243648,
-3.661747819076073, -3.649618198672174, -3.6334890415337426, -3.635232267094215,
-3.6074149671163274, -3.5934981061014266, -3.579582954085555,
-3.564025510623734, -3.5517793023098787, -3.545469970803904,
-3.5256372158055114, -3.520983153663156, -3.501035288863208, -3.484195524911497,
-3.4719173985122267, -3.4565883884270074, -3.441273779533466,
-3.427600130811125, -3.4154020257726554, -3.4154020257726554,
-3.386061970781729, -3.372201565082199, -3.3570241424496867,
-3.3431967408943857, -3.3293821033144297, -3.314345398053944,
-3.3155854756314556, -3.289259973450987, -3.2737111375833665,
-3.2599983587347747, -3.24448897474969, -3.2300849431413154,
-3.2300849431413154, -3.2030938348815763, -3.196381278345486,
-3.174181728449373, -3.160695409690623, -3.143412518747417, -3.1290159033765015,
-3.112667669261392, -3.096316270135664, -3.0915285751066706,
-3.0684129570330456, -3.052092196856459, -3.037823061882639,
-3.0317469431209134, -3.0093503569823725, -2.992492914280152,
-2.979566706674128, -2.964929351665128, -2.9623207487192404,
-2.9349052243560596, -2.916368341714341, -2.900919070973245, -2.887987412063619,
-2.8729522787561605, -2.860342240341454, -2.853694936780197,
-2.8280627902740956, -2.8127476947523986, -2.797309170289222,
-2.7823604900853196, -2.767439229669198, -2.7485306607640916,
-2.7336244248622292, -2.7358126740025526, -2.710311913991678,
-2.6870885969150677, -2.6723903018777984, -2.659507906935932,
-2.6426028622930757, -2.632493222893944, -2.6281495547568863, -2.60029525464943,
-2.579071963520999, -2.5599584013523438, -2.543283353853075, -2.526696757351525,
-2.530834652599799, -2.5011677269306443, -2.490469039506514, -2.492956705896748,
-2.4553593433115117, -2.4345000604346274, -2.418649631410125,
-2.4065887741575964, -2.3907145830234007, -2.3728057611161684,
-2.373482481817198, -2.3445467729484077, -2.32640740146065, -2.3131844080179462,
-2.2944863832444637, -2.2810919866796344, -2.2735091608096254,
-2.248421048004228, -2.2456426542009, -2.2205680797594747, -2.2048250769238384,
-2.185283338599334, -2.1703420258720483, -2.1518311283218123,
-2.149388345831737, -2.1224486292348903, -2.107549210975123,
-2.1098551581144482, -2.077297715154752, -2.05794316297587, -2.0435223075717515,
-2.0258287542357634, -2.021030695101924, -1.9963451319015477,
-1.9758937345333059, -1.9629354239100625, -1.9440265030584396,
-1.9294811581611144, -1.9150101155664505, -1.9193474579221999,
-1.8847676166040288, -1.8832804373611192, -1.8516132650058528,
-1.8343909438109358, -1.8172172727034561, -1.8013612926744211,
-1.7839057602623856, -1.7668587282881247, -1.7709046896564131,
-1.7392950588564933, -1.7207442886087891, -1.7022753968746196,
-1.687734547776687, -1.6704423603730776, -1.665950005733955,
-1.6407031538048593, -1.6297643112344862, -1.6082194291757457,
-1.58933525893743, -1.5676862889751526, -1.5543698848653311,
-1.5409469296246432, -1.5256962131263003, -1.5141638671672728,
-1.491471448989321, -1.4723409110661223, -1.454695287860262,
-1.4396277729201286, -1.424426140991321, -1.4082691672413703,
-1.4069011197124381, -1.3741082240126978, -1.3741082240126978,
-1.3413194763212912, -1.3218681899149698, -1.3117960209504438,
-1.2877637071805461, -1.271285806700163, -1.2693360380655119,
-1.2414990062832891, -1.2236787501799147, -1.2075305396712017,
-1.196056202321214, -1.1722931114046182, -1.1584495376364736,
-1.1604049097459201, -1.1251998212154555, -1.127145254821661,
-1.0939667746112196, -1.0725322965953097, -1.0568317616623135,
-1.0435637883592201, -1.0275092619267312, -1.0098487783372918,
-1.010318576706172, -0.9754643011845819, -0.957287222505 2285,
-0.9420220372683085, -0.9260591531580376, -0.9062939796404088,
-0.8987498960462114, -0.8753080490253671, -0.8668430459793606,
-0.8410827185023795, -0.8227551970338652, -0.8062504803225103,
-0.7927355066523523, -0.7720625212536731, -0.7548615225116874,
-0.7512698548305936, -0.7218559315992382, -0.7081169261927422,
-0.6873050891965011, -0.6682758241806785, -0.6509912156357986,
-0.6381599693487877, -0.6381599693487877, -0.6035400048486522,
-0.6035400048486522, -0.5702208596924025, -0.5526664807070327,
-0.5318450359663531, -0.520176006200697, -0.501413946515456,
-0.49562545963570714, -0.46832962126862043, -0.4464115761777374,
-0.44911543034713963, -0.41909814659381606, -0.39738590401594515,
-0.38247924492673624, -0.3592470863033723, -0.35653010294143384,
-0.32940223595903406, -0.31306382064621086, -0.2913665538558577,
-0.27222378493451127, -0.2584510402332535, -0.25574017672891103,
-0.22601656894545374, -0.20567961762875317, -0.20836334212236848,
-0.17538641392093554, -0.15413200527646292, -0.13816473932896103,
-0.1184552829076625, -0.10035324703877782, -0.09870602062523867,
-0.0710856843245383, -0.050559303282395135, -0.03336842684459443,
-0.01704945555931639, 0.001834971859530271, 0.01995128431460294,
0.02346789977634292, 0.05133719186548716, 0.05743613472341899,
0.08456444302719537, 0.10943984280403093, 0.1277543423540933,
0.14494089794462348, 0.16435143766579477, 0.1717902348511348,
0.1989898209608543, 0.21728993220734297, 0.21497204811208173,
0.2537915190775738, 0.27268133880565193, 0.29400664592616355,
0.30936542852106697, 0.3269368164316252, 0.3269368164316252,
0.36565427508425086, 0.38919661684224416, 0.40286071864431905,
0.42614580158037285, 0.43973719763328456, 0.4455737072346042, 0.480650027408724,
0.48454150109712646, 0.5196448110429285, 0.5333958410595102, 0.5549114885366647,
0.5730638234318912, 0.5929816754896016, 0.6152047606369707, 0.6313569264479292,
0.6313569264479292, 0.6685882269165125, 0.6885381068380857, 0.7096268541589902,
0.728354166348765, 0.7471047539249057, 0.763361578693745, 0.763361578693745,
0.8015164216592982, 0.820978598501558, 0.8378758398259384, 0.8539551383919589,
0.8742153636203044, 0.8715934012796142, 0.9078095565184018, 0.9210820738918611,
0.9514760399486425, 0.9665242772957986, 0.9863041614460754, 1.0040266517756362,
1.0288484481767506, 1.0516108906544563, 1.05440014955545, 1.0912702622161505,
1.1053674902733686, 1.1280206189915047, 1.1424670541484148, 1.1709028782135382,
1.1880095704240248, 1.1880095704240248, 1.2251353610307327, 1.2251353610307327,
1.2639647554753553, 1.2901665423472881, 1.3105583230773106, 1.3309972225427191,
1.350210038792573, 1.3547127104056265, 1.3895695827441086, 1.4088798917698555,
1.4305307502205002, 1.4510619895064145, 1.467277152335229, 1.490359925313339,
1.490359925313339, 1.531975557289195, 1.5347822234690471, 1.5759311656946562,
1.6014969704988844, 1.6163880866403044, 1.6488042829082488, 1.6621226216569411,
1.6887829700315389, 1.6901198676307834, 1.7371973283124058, 1.7480128012439144,
1.754185842115381, 1.8085335601142671, 1.8299149138439135, 1.8510305174465587,
1.8728814412938712, 1.8728814412938712, 1.924168808002698, 1.9218298162658831,
1.971676203176624, 1.9932566259237139, 2.017055171097755, 2.0432729136835306,
2.0673650390637, 2.069675000807777, 2.1223993502831155, 2.1471494849172075,
2.1738758454259752, 2.199686320854443, 2.2308787139428836, 2.252286392292424,
2.254437496942457, 2.3034796795069443, 2.3049608172769287, 2.3541577949583656,
2.3797997421234895, 2.402730391905656, 2.4257830344074285, 2.451940109818534,
2.453014940029663, 2.5004908506819095, 2.529115622942069, 2.555633177734858,
2.584612977099374, 2.604849175419617, 2.6175211884745893, 2.659529083424682,
2.666368407039796, 2.7117157653246, 2.740792830109812, 2.7603480085984073,
2.789459286298334, 2.816998262877869, 2.849365378697212, 2.842747213728028,
2.901231817389677, 2.9286965762103816, 2.95471440349082, 2.9806856960974604,
3.010403386573868, 3.0205529443541623, 3.061618605643668, 3.0757844851425755,
3.114633471180615, 3.1430575050682354, 3.168400131469067, 3.1949213504024536,
3.2238801841471108, 3.2517054438023343, 3.266021584221177, 3.3008663625193693,
3.3314439210536424, 3.3602337256494614, 3.3870166241920026, 3.4154474982247574,
3.4268151697259506, 3.465519301435169, 3.492477553939555, 3.4944951919925833,
3.5435700297225745, 3.570292771819512, 3.598910457323709, 3.6254513088284988,
3.6498066898210255, 3.6803427661033083, 3.680126093902854, 3.730327743564452,
3.7568873415138855, 3.783432265487503, 3.8094324795030197, 3.8357819151524892,
3.8483406255698385, 3.8880747107206086, 3.9181065970185838, 3.915358096984439,
3.9662186703986486, 3.996072149173207, 4.026762792230245, 4.053573237049834,
4.075560655735155, 4.076425620784259, 4.12906986493898, 4.15458426161605,
4.15458426161605, 4.210901932672941, 4.2336840170824654, 4.2593580925849786,
4.281983627119449, 4.297004665101145, 4.331239961456853, 4.3622967290020895,
4.388815732712141, 4.412486302404781, 4.438768080007355, 4.460844415574932,
4.475124987786787, 4.516050675314224, 4.542412700626698, 4.5686845462644134,
4.59483064290047, 4.61873891062957, 4.6454254460788045, 4.64721759646369,
4.70033932667362, 4.698505580272741, 4.751237572910532, 4.77765746338922,
4.804740215099656, 4.830375623621334, 4.85510108471615, 4.8680094101130935,
4.902271883120956, 4.931497304340017, 4.959042625939195, 4.983361474286478,
5.010229123413812, 5.024312699236672, 5.058772815556693, 5.072454859499965,
5.110565964457461, 5.136317218478403, 5.163936652472292, 5.18754443127483,
5.212954775287088, 5.222735513902444, 5.263645839685245, 5.27163272570813,
5.314394863395188, 5.336515585292625, 5.361321723089371, 5.38767115873884,
5.413972969322091, 5.438395912680965, 5.4408005645398045, 5.4885276986960125,
5.4885276986960125, 5.536219340322787, 5.5630999977724525, 5.5908434416156325,
5.613189813441628, 5.639025409829246, 5.646718334298502, 5.6905693962765005,
5.712964447487186, 5.738535580576851, 5.7634791015726865, 5.786421562977982,
5.815517547852859, 5.823885614466716, 5.864483776112307, 5.888914017133541,
5.915906088178919, 5.940219752677559, 5.961778344544222, 5.98883062687237,
5.99943971127939, 6.041066355062132, 6.048181562589321, 6.088539684336785,
6.112818237577017, 6.13808778929538, 6.162910475979256, 6.189659681315188,
6.216008196402982, 6.224336008012897, 6.261836803102101, 6.287448481019581,
6.3155294731186125, 6.338527847543341, 6.362590381428089, 6.389124340562046,
6.390302813238962, 6.444622845807762, 6.444622845807762, 6.493255869954863,
6.521032593201532, 6.5458096627696625, 6.572895548028766, 6.599993364813137,
6.624971080170097, 6.624971080170097, 6.673155204062807, 6.703694105713167,
6.72678348948827, 6.757590690022374, 6.780661723337978, 6.790387529334779,
6.832941563980229, 6.848713936006344, 6.885848963663247, 6.915762287525525,
6.942434976425172, 6.967901167939738, 6.993973968604203, 7.019986717153916,
7.036041954184611, 7.073695173780156, 7.103064317595782, 7.129363876634519,
7.155742878352021, 7.182164855966869, 7.209413239914839, 7.222582656979528,
7.265377280427863, 7.293305786462816, 7.319180286818667, 7.347332854439515,
7.375583054784423, 7.406402300729448, 7.406689034505202, 7.46302589045856,
7.460062710129594, 7.516867468642935, 7.545340784084253, 7.574128086602283,
7.602596242134137, 7.631611455993173, 7.6421043105839175, 7.683660981366689,
7.70402131442898, 7.743809282322825, 7.773303682509493, 7.800748740307002,
7.830071222606965, 7.858714034029898, 7.88746871961638, 7.884747655387198,
7.946366518329388, 7.973943258185091, 8.003063479554987, 8.031500497286377,
8.061429842636748, 8.060462363442575, 8.124415053451981, 8.12233970521826,
8.18142724345575, 8.18352868186184, 8.240690037911346, 8.272490598426048,
8.304379408548913, 8.336427887372771, 8.366707223905248, 8.370818914529695,
8.431665739635184, 8.462768198458644, 8.498320710119433, 8.529769160554794,
8.561875640359892, 8.592855010200248, 8.594924331971008, 8.662138770700013,
8.662138770700013, 8.722496859053782, 8.758339202947656, 8.789581853742142,
8.821793280034127, 8.853244286996793, 8.856256958834066, 8.916253157483586,
8.950348288954592, 8.98453226112305, 9.015636511824516, 9.046469190470564,
9.080667381359893, 9.082212960345764, 9.144242158339601, 9.175391646335395,
9.208685303974013, 9.244265672419841, 9.278945640078204, 9.291727448867205,
9.339120823376314, 9.358694336449648, 9.410671156881154, 9.442264004951392,
9.474625131790514, 9.508583922937948, 9.538603740362202, 9.573268315632818,
9.573268315632818, 9.64132996084776, 9.64132996084776, 9.709047067782903,
9.739889674276347, 9.772468309539256, 9.807807463903476, 9.840554603160836,
9.855819725390885, 9.905885875251494, 9.938294042314377, 9.97086618609808,
10.000917762803613, 10.034958526897762, 10.065703367577196, 10.067807869773093,
10.132829833542797, 10.152265215326887, 10.19846023024756, 10.228020405805157,
10.260774466280775, 10.293072812319112, 10.326265236474077, 10.347208083385489,
10.394730931179296, 10.40855783122471, 10.457783950833562, 10.488196003467245,
10.518931031055672, 10.551722646358701, 10.58453045972127, 10.605432295431218,
10.605563723351008, 10.685296777924265, 10.711781167664494, 10.746669078364455,
10.778985262495281, 10.814580767298088, 10.843872469060008, 10.846167329634447,
10.903378344138751, 10.938587708321942, 10.9689090622289, 11.006321525996745,
11.038475378573446, 11.055561537592967, 11.105839076559658, 11.124450955124932,
11.166211455138601, 11.198191206472897, 11.231566804313703, 11.261570818290624,
11.293983582495514, 11.32288435436737, 11.345374402352327, 11.39142750191626,
11.421395882497421, 11.45118316818935, 11.482894588474544, 11.514510556167012,
11.550112630306725, 11.550112630306725, 11.608847062735734, 11.613056395600582,
11.668872947006133, 11.70447897798326, 11.731077790189836, 11.76629254666268,
11.79453680422062, 11.830252081727298, 11.82920345932815, 11.890301301868618,
11.889410773272687, 11.950860492656538, 11.985323810623848, 12.015815568336675,
12.043664787161529, 12.074181210663543, 12.074181210663543, 12.137455204887734,
12.16611913341453, 12.197584519996758, 12.227512264749814, 12.261447966608536,
12.276434210652484, 12.322398500358082, 12.35263388462124, 12.35263388462124,
12.415451286862528, 12.447305615742767, 12.476518213210376, 12.50828547937545,
12.535616853876329, 12.538186724733295, 12.602218029380042, 12.635216523179789,
12.663729116029884, 12.694065690429918, 12.72569622370909, 12.742043943156526,
12.791802928189284, 12.820759861996757, 12.821112969326714, 12.882943293073772]
```

[ ]:

```
time_1 = np.arange(len(theta))/frame_rate
popt_1, _ = sp.optimize.curve_fit(quad, time_1, theta)
popt_1
```

[ ]:
```
array([ 0.30196946, -2.03139018, -1.44260368])
```

[ ]:
```
plt.plot(time_1, theta)
plt.plot(time_1, quad(time_1, *popt_1))
plt.xlabel(r"Temps $[s]$")
plt.ylabel(r"$\theta$ $[rad]$")
plt.savefig("p70_t.png")
```

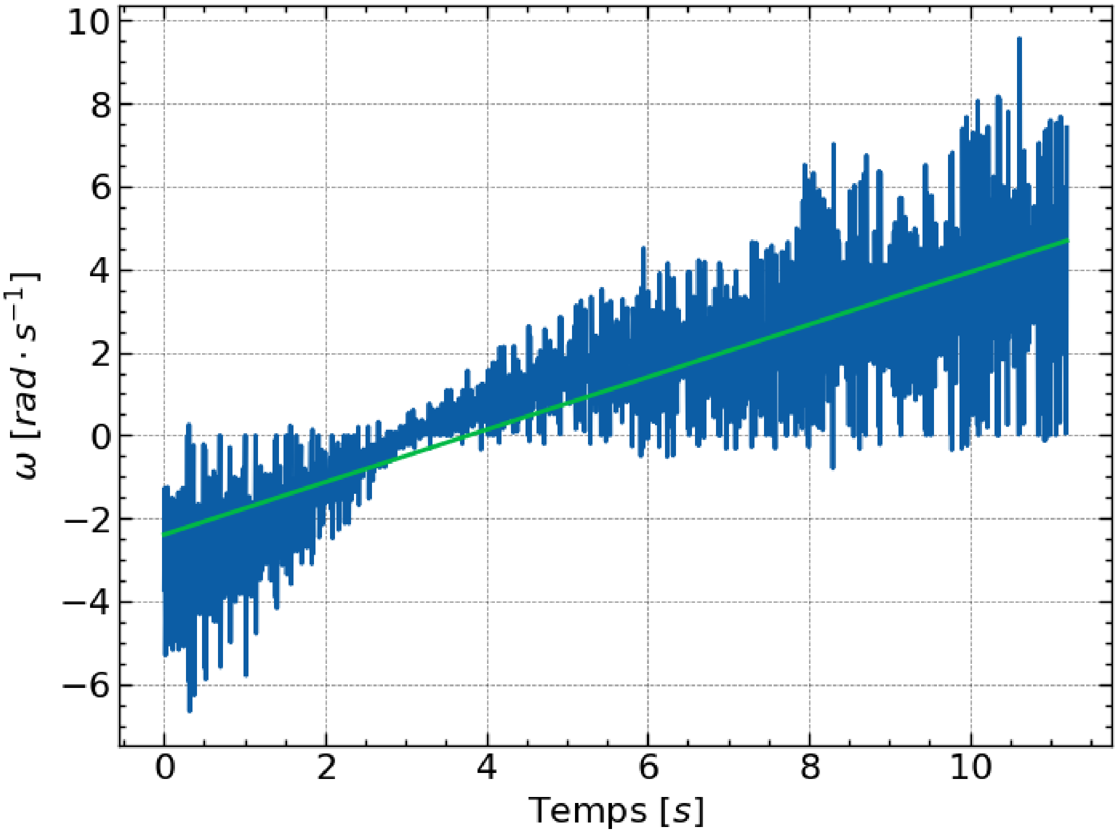


[ ]:
```
omega = []

for i in range(len(theta)-1):
    omega.append((theta[i+1]-theta[i])/(1/frame_rate))
```

[ ]:
```
time_2 = np.arange(len(omega))/frame_rate
popt_2, _ = sp.optimize.curve_fit(lin, time_2, omega)
popt_2
```

[ ]:
```
array([ 0.63267447, -2.39356945])
```

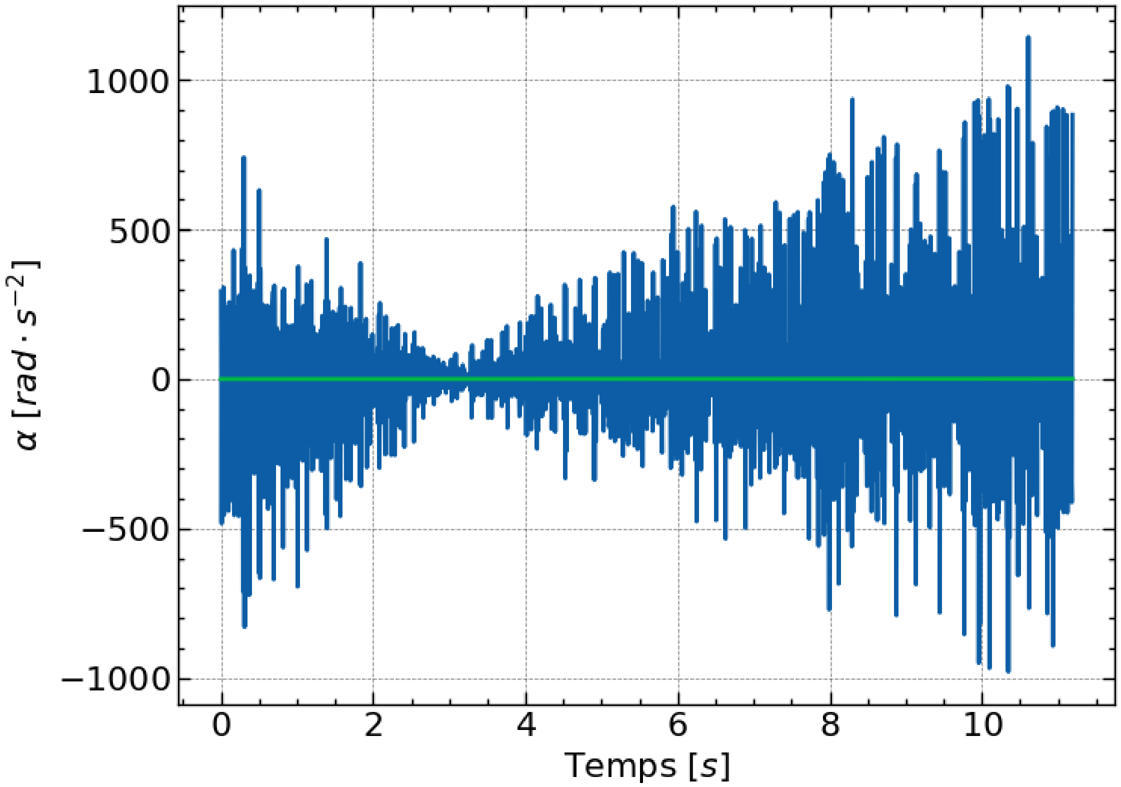
[ ]:
```
plt.plot(time_2, omega)
plt.plot(time_2, lin(time_2, *popt_2))
plt.xlabel(r"Temps $[s]$")
plt.ylabel(r"$\omega$ $[rad \cdot s^{-1}]$")
plt.savefig("p70_o.png")
```



[ ]:

```
alpha = []

for i in range(len(omega)-1):
    alpha.append((omega[i+1]-omega[i])/(1/frame_rate))

time_3 = np.arange(len(alpha))/frame_rate
popt_3, _ = sp.optimize.curve_fit(lin, time_3, alpha)
print(popt_3)

plt.plot(np.arange(len(alpha))/frame_rate, alpha)
plt.plot(np.arange(len(alpha))/frame_rate, lin(time_3, *popt_3))
plt.xlabel(r"Temps $[s]$")
plt.ylabel(r"$\alpha$ $[rad \cdot s^{-2}]$")
plt.savefig("p70_a.png")
```

```
[0.06707486 0.61923022]
```



[ ]:

```
acc_quad = popt_1[0]*2
acc_lin = popt_2[0]
print(acc_quad, acc_lin)
acc = (acc_quad + acc_lin)/2
Mom = 9.81 * m70 * r_p
I_p70 = Mom/acc
print(I_p70)
```

```
0.6039389114987118 0.6326744702267937
0.016659208370569663
```

1.4 Vidéo m70

[ ]:

```
video_path = "m70.mp4"
cap = cv2.VideoCapture(video_path)
```

[ ]:

```
frame_rate = cap.get(cv2.CAP_PROP_FPS)
print("Frame rate: ", frame_rate)
```

```
Frame rate:  30.0
```

[ ]:

```
frames = []
while True:
    ret, frame = cap.read()
    if not ret:
        break
    frames.append(frame)
```

[ ]:

```
frame_rate = 120
print("Frame rate: ", frame_rate)
```

```
Frame rate:  120
```

[ ]:

```
def disp_f(num):
    plt.imshow(frames[num])

widgets.interact(disp_f, num=widgets.IntSlider(min=0, max=len(frames)-1,␣
 ↪step=1))
```

```
interactive(children=(IntSlider(value=0, description='num', max=1587),␣
 ↪Output()), _dom_classes=('widget-intera…
```

[ ]:

```
<function __main__.disp_f(num)>
```

[ ]:

```
analyse = frames[755:1395]
```

[ ]:

```
def pos_track(x,y):
    image = analyse[0].copy()
    w, h = 50, 50
    cv2.rectangle(image, (x, y), (x + w, y + h), (255, 0, 0), 4)
    plt.imshow(image)

widgets.interact(pos_track, x=widgets.IntSlider(min=0, max=len(analyse[0]),␣
 ↪step=1), y=widgets.IntSlider(min=0, max=len(analyse[0][0]), step=1))
```

```
interactive(children=(IntSlider(value=0, description='x', max=1080),␣
 ↪IntSlider(value=0, description='y', max=1…
```

[ ]:

```
<function __main__.pos_track(x, y)>
```

[ ]:

```
tracker = cv2.TrackerCSRT_create()

track_frames = analyse.copy()

init_trac_f = track_frames[0]
roi = (1002, 310, 50, 50)

tracker.init(init_trac_f, roi)
```

```
results = []
frame_tracked = []
for i in range(1, len(track_frames)):
    # Read the next frame
    frame = track_frames[i]

    # Update the tracker
    success, roi = tracker.update(frame)

    if success:
        # Draw a bounding box around the tracked object
        (x, y, w, h) = tuple(map(int, roi))
        results += [(x,y,w,h)]
        cv2.rectangle(frame, (x, y), (x + w, y + h), (255, 0, 0), 4)
        frame_tracked.append(frame)
```

[ ]:

```
def disp_t(num):
    plt.imshow(frame_tracked[num])

widgets.interact(disp_t, num=widgets.IntSlider(min=0, max=len(frame_tracked)-1,␣
 ↪step=1))
```

```
interactive(children=(IntSlider(value=0, description='num', max=638), Output()),␣
 ↪_dom_classes=('widget-interac…
```

[ ]:

```
<function __main__.disp_t(num)>
```

[ ]:

```
good_results = results
```

[ ]:

```
center_of_rot = (725, 535, 50, 50)
```

[ ]:

```
def pos_from_case(case):
    x,y,w,h = case
    return (x + (x+w))/2, (y + (y+h))/2
```

[ ]:

```
pos_c = pos_from_case(center_of_rot)
pos_r = [pos_from_case(i) for i in good_results]
```

[ ]:

```
theta = []
cur_theta = 0

prev_dir = [pos_r[0][0]-pos_c[0], pos_r[0][1]-pos_c[1]]
for i in range(len(pos_r)):
    next_dir = [pos_r[i][0]-pos_c[0], pos_r[i][1]-pos_c[1]]
    cur_theta += (np.arctan2(next_dir[1], next_dir[0]) - np.
 ↪arctan2(prev_dir[1], prev_dir[0]))
```

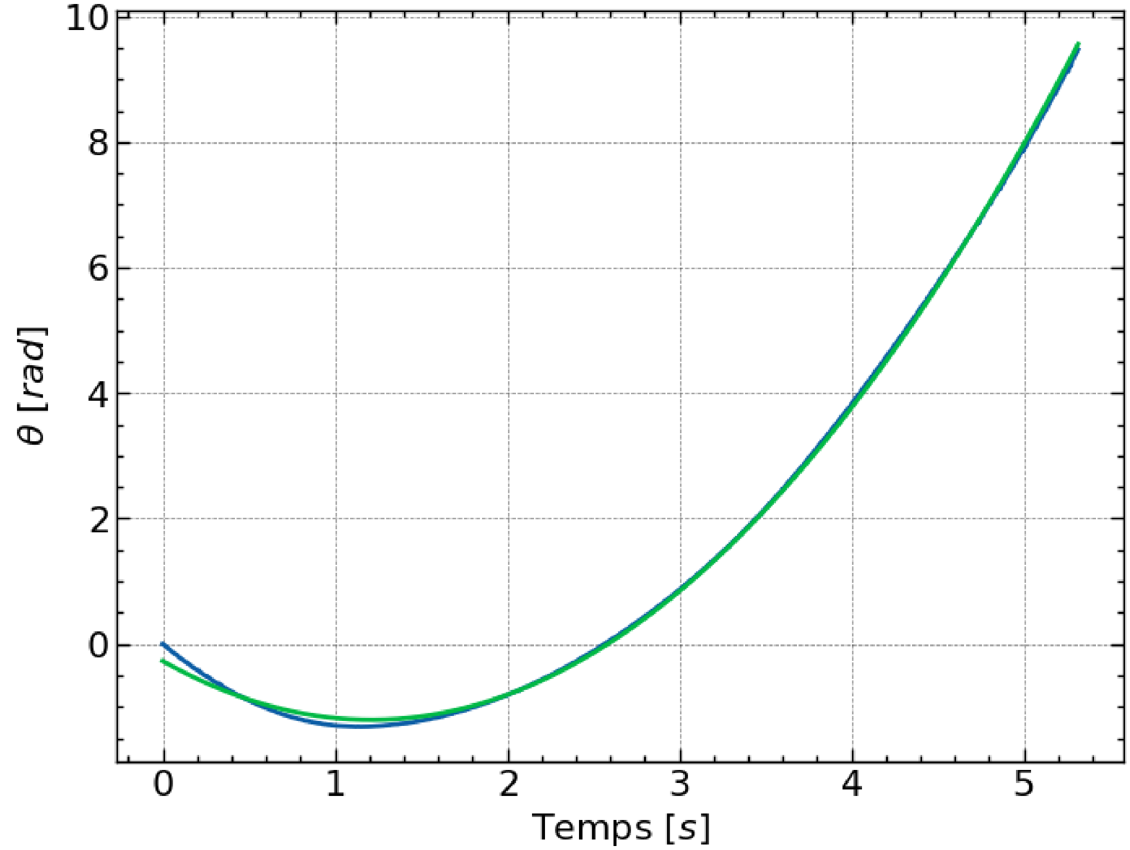
```
    if np.abs(np.arctan2(next_dir[1], next_dir[0]) - np.arctan2(prev_dir[1],␣
 ↪prev_dir[0])) > np.pi:
        cur_theta -= 2*np.pi*np.sign(np.arctan2(next_dir[1], next_dir[0]) - np.
 ↪arctan2(prev_dir[1], prev_dir[0]))
    theta.append(cur_theta)
    prev_dir = next_dir

print(theta)
```

```
[0.0, -0.010025296500136505, -0.03769663497479936, -0.04759916896247118,
-0.0753790716637942, -0.09122026243017767, -0.11305037484520375,
-0.1307928407217125, -0.15066260033062517, -0.16463560309871628,
-0.1725327890724142, -0.20250429057724917, -0.22018389985061337,
-0.23535253400356493, -0.2530427887949934, -0.26780941363576927,
-0.2873518033709801, -0.2959557672839802, -0.31773497135077733,
-0.3288084265610671, -0.35345385388079387, -0.3693123401815608,
-0.38495186184993657, -0.401359275938594, -0.4171437278535861,
-0.416493732530983, -0.4480167899621301, -0.46280993367532086,
-0.4817639454518263, -0.49700273724435284, -0.510415931738547,
-0.5122383571943714, -0.5390355835607121, -0.5426437674753151,
-0.5695707339327256, -0.585748074587356, -0.6001875333241435,
-0.6136433293620032, -0.6263893695377152, -0.6270757001444605,
-0.6543219817197063, -0.6543219817197063, -0.6811209270529175,
-0.6945200649092911, -0.7074045150508463, -0.721928878503288,
-0.7348291603874391, -0.741714644792349, -0.7590746069638423,
-0.7691629226095671, -0.7864712031342547, -0.7963167004279799,
-0.8120898078819523, -0.8205048368787073, -0.8305847849578398,
-0.8448956502504885, -0.8448956502504885, -0.8678207665665885,
-0.8792736784751818, -0.8907325192334778, -0.9007979147189162,
-0.9108861044903178, -0.91525267269145, -0.9338122750293194,
-0.9395343909502756, -0.9558548150748931, -0.9626368268306205,
-0.9727747739072687, -0.9816527848200656, -0.9930731623784341,
-1.0023712287594406, -1.003610241238998, -1.020704858433339,
-1.0235456660392745, -1.039848322225643, -1.046018214805597,
-1.0550504152246674, -1.0635511633287085, -1.0743056754280067,
-1.0799574520967712, -1.0929411436560874, -1.098555265246978,
-1.1083011807876824, -1.1167051488028241, -1.1230067999237818,
-1.1303510983471399, -1.1331373014755075, -1.1440202346168014,
-1.1521572001772737, -1.1604758643613935, -1.1658779200137412,
-1.1722662023351575, -1.1787306826730075, -1.1796086455729726,
-1.1896736420780163, -1.194256607574125, -1.2016025105909833,
-1.2061890553280614, -1.2116568551455804, -1.220809112135088,
-1.2253835536601714, -1.2253835536601714, -1.235560550141119,
-1.235560550141119, -1.2430763483259852, -1.2474188312540218,
-1.2511842787615959, -1.2549619186439422, -1.259546641155953,
-1.2641285395781667, -1.265245007015822, -1.2732831065535646,
-1.2770964043053765, -1.2790074131238136, -1.2824241121918643,
-1.2862481531194663, -1.290083565587369, -1.288164441779917, -1.295384167552998,
-1.2965830255302169, -1.299229394326035, -1.303085677219769, -1.303085677219769,
-1.3057247291523866, -1.306952948801268, -1.3083573234384809,
-1.3095909727264674, -1.3122224787986285, -1.3122224787986285,
-1.3134680403097434, -1.314159084184904, -1.3160983627801994,
-1.3160983627801994, -1.3180403055066914, -1.3200777701532416,
-1.3187220956722001, -1.3160983627801994, -1.3187220956722001,
-1.3187220956722001, -1.3187220956722001, -1.3187220956722001,
-1.3187220956722001, -1.3180403055066914, -1.3180403055066914,
-1.3160983627801994, -1.3148474548335094, -1.3115281579476359,
-1.3129141174504788, -1.3122224787986285, -1.310288555593981,
-1.3095909727264674, -1.306952948801268, -1.3057247291523866,
-1.303085677219769, -1.3018693794210177, -1.2965830255302169,
-1.2980249916775657, -1.2927370206902968, -1.2939302870518266,
-1.288164441779917, -1.2855044011964973, -1.283587843704577,
-1.2816741523584203, -1.2809213188007638, -1.2797633347187047,
-1.274423337653122, -1.272515096441923, -1.2687074243219845,
-1.2660279927832296, -1.2633425556118278, -1.2614430981211804,
-1.2603385733897001, -1.2511842787615959, -1.2511842787615959,
-1.2466033353273218, -1.2430763483259852, -1.23826788192699, -1.235560550141119,
-1.2318214488221955, -1.229109243802525, -1.2218047595232402,
-1.217216766083173, -1.2135069102985243, -1.208925584404175,
-1.2043435658456263, -1.2016025105909833, -1.1988269062037242,
-1.190595277531082, -1.1841868844079904, -1.1805460171421205,
-1.174081890932753, -1.1704539717492106, -1.1668398925140897,
-1.1658779200137412, -1.1521572001772737, -1.1549345990054347,
-1.1430181556376404, -1.1338873638400562, -1.1321174246306558,
-1.1268292842148344, -1.1191653223066338, -1.1128534887258634,
-1.1146043028003867, -1.099207226020343, -1.0918515385910252,
-1.0884096406747996, -1.0821729219361333, -1.0726008782166667,
-1.0708999095970833, -1.0618623612462454, -1.0556727443149687,
-1.0556727443149687, -1.0415110275469954, -1.0329802869399332,
-1.0268306096241724, -1.0182845190064553, -1.0121679558688677,
-1.0076949624581566, -0.9975293246761274, -0.9943214057432992,
-0.9829209128037111, -0.9756299549290119, -0.9683488399029543,
-0.9569187898003986, -0.9496500093906233, -0.9467895817473275,
-0.9366738739940808, -0.925221449362541, -0.9194896732499171,
-0.9122608041948079, -0.9050483151922353, -0.8935965241390023,
-0.8879175728437607, -0.8749820357588534, -0.8678125799778584,
-0.8563522341590539, -0.8477592791594597, -0.8377856200780678,
-0.8305847849578398, -0.8292138408468719, -0.813433920550005,
-0.8120898078819523, -0.7919314627042706, -0.784928804643878,
-0.7764023277254166, -0.7647539994965686, -0.7562375963791677,
-0.7480865737041624, -0.7480865737041624, -0.7279763419893492,
-0.7162965983674754, -0.7045920173862308, -0.6956750148223576,
-0.6849809724319678, -0.6788626410184648, -0.6732823417993503,
-0.6560138635067596, -0.6504704954506022, -0.6325909281719497,
-0.6208750168159709, -0.6119070096855335, -0.6001875333241435,
-0.5919886828663969, -0.5865440031233651, -0.5677930332742596,
-0.5534116221274048, -0.5453495449874486, -0.5337363800418637,
-0.5211815597012321, -0.5122383571943714, -0.5122383571943714,
-0.4880706872914309, -0.4880706872914309, -0.46393082992326007,
-0.45130513731793087, -0.43866530487653743, -0.4247949211419475,
-0.41460590529962027, -0.4057692592689399, -0.3918607862891501,
-0.38055476541526145, -0.36659722074295664, -0.35345385388079387,
-0.3394546571288346, -0.33702485200053944, -0.3143177227336924,
-0.301791190170156, -0.3022193300347411, -0.2795527477507158,
-0.26548949025576607, -0.25141105537634045, -0.2373230061151912,
-0.21928073586622154, -0.20716254994868055, -0.208894723066826,
-0.18277405141404468, -0.16875067668943744, -0.15672848588439525,
-0.14298452783822968, -0.13106445255416033, -0.13106445255416033,
-0.10118296997057119, -0.09323185690928382, -0.07339910914554981,
-0.05750910165836165, -0.04363715915319433, -0.02570815997758591,
-0.011853333284198908, 0.0019733125688488062, 0.012021587827255709,
0.030204123808249772, 0.045702552492658755, 0.06194387096666798,
0.07564132207297725, 0.0915681128830339, 0.10590472134523166,
0.10358889648100456, 0.13231509417916598, 0.13574567284050865,
0.16498479587534942, 0.18460991827392292, 0.19574100247279402,
0.21336437335660535, 0.2277319624518377, 0.24272073476655842,
0.2496197139103341, 0.27534707837901834, 0.28971198888158595,
0.31027137522170567, 0.3253203409875523, 0.33856589328991277, 0.356247585520268,
0.3650943140648488, 0.3889775522125583, 0.4030542128046228, 0.4225868151446763,
0.43509487660521773, 0.45886729473769405, 0.4714315994734414,
0.4741592074570249, 0.5071160275877127, 0.5071160275877127, 0.540027175876042,
0.5570351314775093, 0.5752426877193995, 0.5919387461307727, 0.6102205393214392,
0.628511102943993, 0.6300088036684892, 0.6650696700713821, 0.6818876260817047,
0.6987198246589913, 0.7183492242096265, 0.735265689187887, 0.735265689187887,
0.771725739996932, 0.7803408095148188, 0.8086017367006371, 0.8260487728439738,
0.8460074745122527, 0.8627214010472624, 0.8826144253937017, 0.9024753555326391,
0.9024753555326391, 0.9406526730915704, 0.9604490039070398, 0.9782246979609621,
0.9968518585142642, 1.017404723485178, 1.0272404587672954, 1.0567385610970979,
1.0630274594756148, 1.0945822840302315, 1.1161829308694886, 1.1340091457487662,
1.1542425125528593, 1.1732895557400849, 1.193674363365505, 1.2050124690692952,
1.2342150524792808, 1.2585671607583984, 1.2766726540289886, 1.2995513546752053,
1.3200376601497408, 1.3302718461282566, 1.360412717451374, 1.3706399831262148,
1.4045580261234092, 1.4268794320188363, 1.447073998521926, 1.4673734182720541,
1.487540877276079, 1.5118263057960362, 1.5118263057960362, 1.554652661257577,
1.554652661257577, 1.599539289494368, 1.6179338926526747, 1.6409535820109387,
1.6657344534132916, 1.6865078180747548, 1.688914477268468, 1.7339743628811162,
1.743359460356773, 1.7807017801875034, 1.804030908283038, 1.8260594865011635,
1.8519554526274367, 1.8713400442984338, 1.88582989887542, 1.9225160953668774,
1.947327652317657, 1.968463271767633, 1.9924152927383096, 2.019224780182376,
2.039740521612914, 2.0404073694047895, 2.090146100096074, 2.0872719066371923,
2.1369078990452457, 2.1622005335028964, 2.188747229578862, 2.2121589292248394,
2.238667869505308, 2.2503897754063056, 2.2885599792316844, 2.3120679745820043,
2.33865754321061, 2.3653079551299947, 2.389056376153891, 2.4149038537018646,
2.4207547833100875, 2.4717885586330963, 2.4688903581291015, 2.5241636328396035,
2.546558645994374, 2.5767191908930016, 2.6020547017636053, 2.628465272912508,
2.6566682501875776, 2.6566682501875776, 2.709884249812653, 2.7365636924921826,
2.7632402870816035, 2.791357401856219, 2.81698100944159, 2.832836314197377,
2.8709649653423153, 2.8883816364095285, 2.930906119969541, 2.958544598397138,
2.9819820281703446, 3.0138755642414514, 3.04142039940338, 3.0690510798280743,
3.0860281576122555, 3.12868098578622, 3.1564046691366943, 3.1840925540306655,
3.2099317053781755, 3.2382608522812024, 3.2708258010958433, 3.2816761287106297,
3.3234986710336574, 3.355754798082713, 3.3544988089002934, 3.413834835008923,
3.443554531332987, 3.473238825391549, 3.502835652409067, 3.512348873054221,
3.559062720699171, 3.5876052502976816, 3.618301282324501, 3.6460885666729896,
3.67680998735192, 3.707470190995301, 3.707066456701635, 3.767938717002415,
3.767938717002415, 3.82517123376607, 3.8552440871059015, 3.885292488352298,
3.9184829700052357, 3.9454101617088577, 3.978521275986781, 3.978521275986781,
4.039106638585338, 4.0696315055025405, 4.100099594034271, 4.131306211322791,
4.161576621227148, 4.1766311445361435, 4.225821543375504, 4.256762734264388,
4.257966012284832, 4.317965849890165, 4.348521232501415, 4.379077153381019,
4.412070868008681, 4.4268389468756, 4.47366398904102, 4.493178970878064,
4.539285422200754, 4.570683845714788, 4.602166391169247, 4.6333332281534405,
4.6645138255003875, 4.681110742508874, 4.728840172583103, 4.759934259996397,
4.790900607859848, 4.822479693370033, 4.8556058814247685, 4.889568641959182,
4.889568641959182, 4.956016473302446, 4.953408173221272, 5.018273761522647,
5.049495878580233, 5.084358400975861, 5.1173073871658445, 5.15106788254893,
5.165776203013022, 5.213982540683421, 5.231494560643485, 5.27872376315676,
5.313551105105621, 5.344984314619277, 5.3795208098689455, 5.413938019718882,
5.445399687101518, 5.445399687101518, 5.514022384570019, 5.545530146914762,
5.576954000214563, 5.614936479916626, 5.646778360462933, 5.663333955211107,
5.714458567360603, 5.72886717656779, 5.781713343846677, 5.814800830922142,
5.850162138196814, 5.8805825630821875, 5.913550382509998, 5.949041544099209,
5.966472279543395, 6.0173381659905, 6.033739853188729, 6.086148722070005,
6.122202216405676, 6.156097276018486, 6.189953450270303, 6.22573138706846,
6.239713861837122, 6.293802901466239, 6.329406819347305, 6.363708054683399,
6.395690385278071, 6.429904191838335, 6.4688524192662245, 6.483844915588984,
6.535328821143258, 6.571777321679763, 6.608270326368918, 6.64112973031543,
6.676649721902931, 6.712131673672946, 6.712131673672946, 6.784258008956693,
6.801316969361479, 6.855294673652057, 6.891880874774082, 6.9283520593037595,
6.967877186183653, 7.001585802114374, 7.018550060410074, 7.0719285167644585,
7.108880579990624, 7.143576664769016, 7.1806992921751345, 7.218354614815669,
7.2559865609937795, 7.256748117771936, 7.328326718155449, 7.346212766655202,
7.4019595303086, 7.439976184949523, 7.478013002802479, 7.517400359658868,
7.553528675176727, 7.590648574003646, 7.608852936824902, 7.66762762166731,
7.706031037842983, 7.74239217846978, 7.780822514264792, 7.821391419592371,
7.860281745865388, 7.860281745865388, 7.938537515716898, 7.938537515716898,
8.013680558849567, 8.052332592889018, 8.09239923546005, 8.133959740728502,
8.175477932806098, 8.214888044327413, 8.232349504723508, 8.293216056566399,
8.334956354617532, 8.37333140727566, 8.414685415635251, 8.455743933219598,
8.496725689505945, 8.517297831901665, 8.579131648559047, 8.62027721607328,
8.66214717202883, 8.703500099230768, 8.746269185722042, 8.764263352309305,
8.826079720722387, 8.850527697473105, 8.910611042640646, 8.955868075792264,
8.998455318135617, 9.04239291034509, 9.08529378322086, 9.128404889294853,
9.125887867663316, 9.212257611635987, 9.212257611635987, 9.301300250886836,
9.343744481497449, 9.386173427325184, 9.428812215510824, 9.473876621809461]
```

```
[ ]: time_1 = np.arange(len(theta))/frame_rate
     popt_1, _ = sp.optimize.curve_fit(quad, time_1, theta)
     popt_1
```

```
[ ]: array([ 0.63896602, -1.54767844, -0.27137422])
```

```
[ ]: plt.plot(time_1, theta)
     plt.plot(time_1, quad(time_1, *popt_1))
     plt.xlabel(r"Temps $[s]$")
     plt.ylabel(r"$\theta$ $[rad]$")
     plt.savefig("m70_t.png")
```



```
[ ]: omega = []

     for i in range(len(theta)-1):
```

```
    omega.append((theta[i+1]-theta[i])/(1/frame_rate))
```

[ ]:

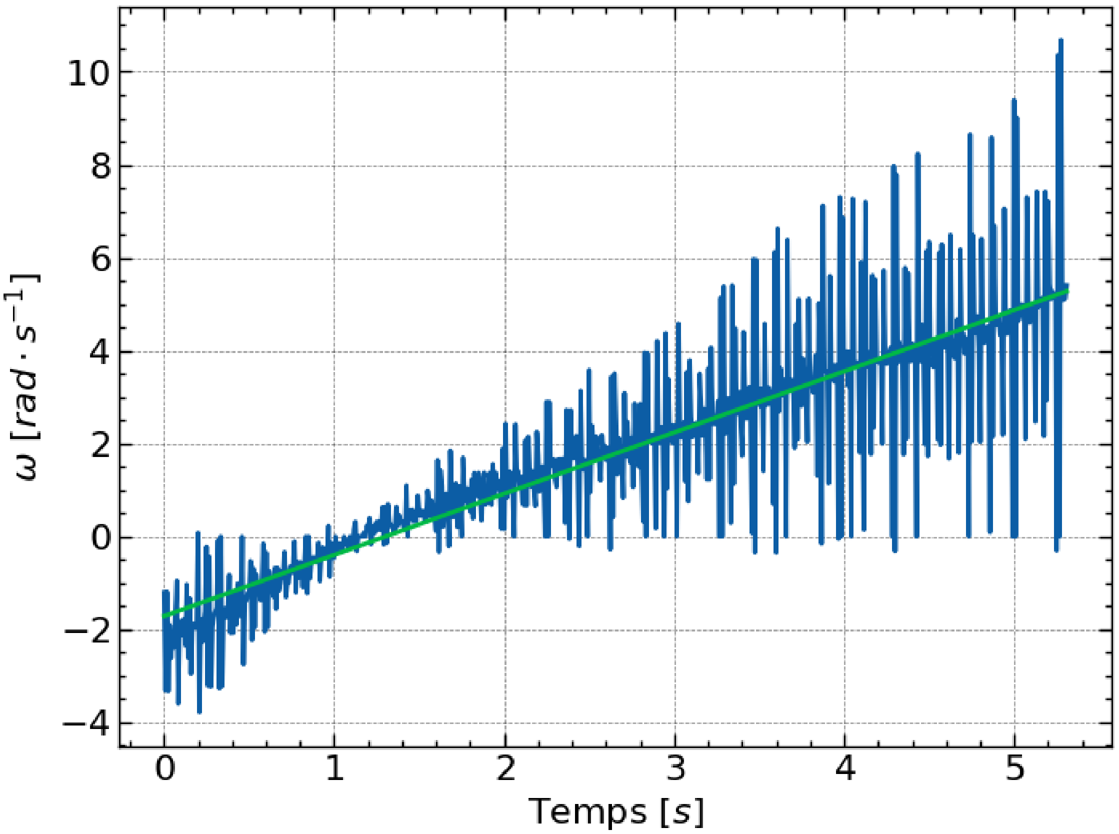
```
time_2 = np.arange(len(omega))/frame_rate
popt_2, _ = sp.optimize.curve_fit(lin, time_2, omega)
popt_2
```

[ ]:

```
array([ 1.31694293, -1.71346566])
```

[ ]:

```
plt.plot(time_2, omega)
plt.plot(time_2, lin(time_2, *popt_2))
plt.xlabel(r"Temps $[s]$")
plt.ylabel(r"$\omega$ $[rad \cdot s^{-1}]$")
plt.savefig("m70_o.png")
```

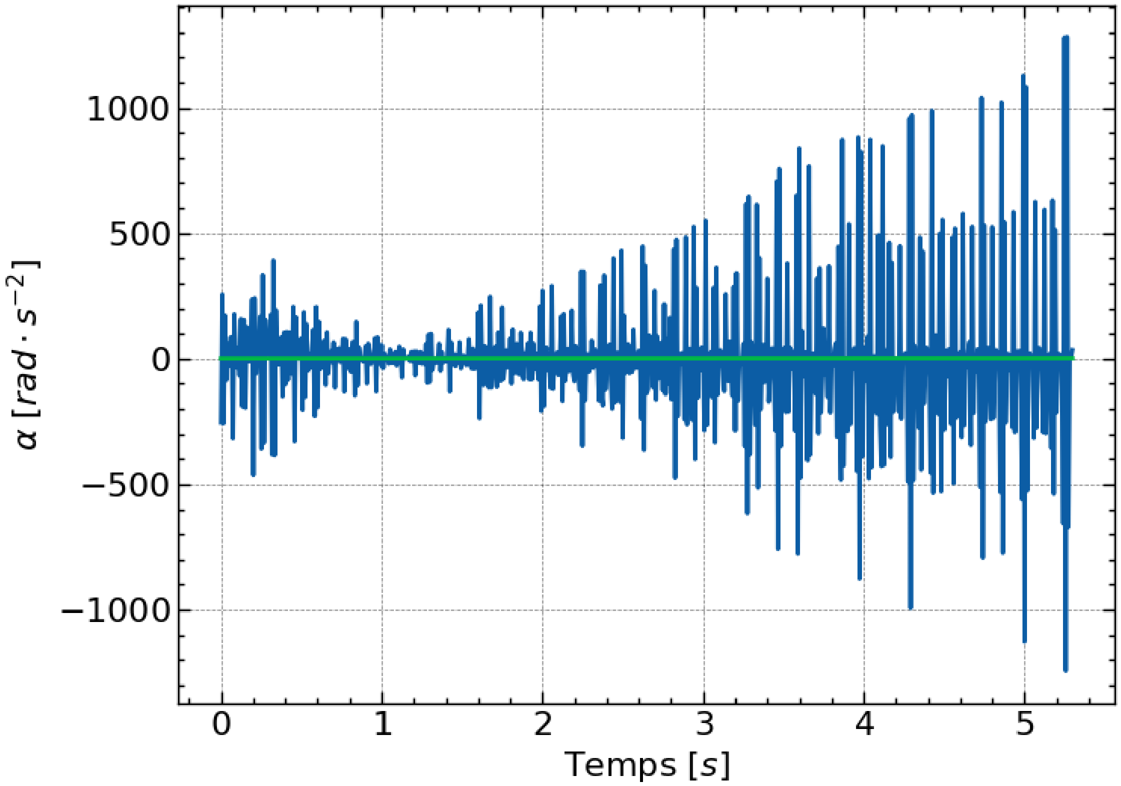


[ ]:

```
alpha = []

for i in range(len(omega)-1):
    alpha.append((omega[i+1]-omega[i])/(1/frame_rate))

time_3 = np.arange(len(alpha))/frame_rate
popt_3, _ = sp.optimize.curve_fit(lin, time_3, alpha)
```

```
print(popt_3)

plt.plot(np.arange(len(alpha))/frame_rate, alpha)
plt.plot(np.arange(len(alpha))/frame_rate, lin(time_3, *popt_3))
plt.xlabel(r"Temps $[s]$")
plt.ylabel(r"$\alpha$ $[rad \cdot s^{-2}]$")
plt.savefig("m70_a.png")
```

```
[0.13666336 0.88319805]
```



[ ]:

```
acc_quad = popt_1[0]*2
acc_lin = popt_2[0]
print(acc_quad, acc_lin)
acc = (acc_quad + acc_lin)/2
Mom = 9.81 * m70 * r_m
I_m70 = Mom/acc
print(I_m70)
```

```
1.2779320473292384 1.3169429297856026
0.015878221634327522
```

1.5 Vidéo g70

```
[ ]: video_path = "g70.mp4"
     cap = cv2.VideoCapture(video_path)
```

```
[ ]: frame_rate = cap.get(cv2.CAP_PROP_FPS)
     print("Frame rate: ", frame_rate)
```

```
Frame rate:  30.0
```

```
[ ]: frames = []
     while True:
         ret, frame = cap.read()
         if not ret:
             break
         frames.append(frame)
```

```
[ ]: frame_rate = 120
     print("Frame rate: ", frame_rate)
```

```
Frame rate:  120
```

```
[ ]: def disp_f(num):
         plt.imshow(frames[num])

     widgets.interact(disp_f, num=widgets.IntSlider(min=0, max=len(frames)-1,␣
      ↪step=1))
```

```
interactive(children=(IntSlider(value=0, description='num', max=2152),␣
 ↪Output()), _dom_classes=('widget-intera…
```

```
[ ]: <function __main__.disp_f(num)>
```

```
[ ]: analyse = frames[903:2014]
```

```
[ ]: def pos_track(x,y):
         image = analyse[0].copy()
         w, h = 50, 50
         cv2.rectangle(image, (x, y), (x + w, y + h), (255, 0, 0), 4)
         plt.imshow(image)

     widgets.interact(pos_track, x=widgets.IntSlider(min=0, max=len(analyse[0]),␣
      ↪step=1), y=widgets.IntSlider(min=0, max=len(analyse[0][0]), step=1))
```

```
interactive(children=(IntSlider(value=0, description='x', max=1080),␣
 ↪IntSlider(value=0, description='y', max=1…
```

```
[ ]: <function __main__.pos_track(x, y)>
[ ]: tracker = cv2.TrackerCSRT_create()

     track_frames = analyse.copy()

     init_trac_f = track_frames[0]
     roi = (1054, 635, 50, 50)

     tracker.init(init_trac_f, roi)

     results = []
     frame_tracked = []
     for i in range(1, len(track_frames)):
         # Read the next frame
         frame = track_frames[i]

         # Update the tracker
         success, roi = tracker.update(frame)

         if success:
             # Draw a bounding box around the tracked object
             (x, y, w, h) = tuple(map(int, roi))
             results += [(x,y,w,h)]
             cv2.rectangle(frame, (x, y), (x + w, y + h), (255, 0, 0), 4)
             frame_tracked.append(frame)
```

```
[ ]: def disp_t(num):
         plt.imshow(frame_tracked[num])

     widgets.interact(disp_t, num=widgets.IntSlider(min=0, max=len(frame_tracked)-1,␣
      ↪step=1))
```

```
interactive(children=(IntSlider(value=0, description='num', max=1109),␣
 ↪Output()), _dom_classes=('widget-intera…
```

```
[ ]: <function __main__.disp_t(num)>
```

```
[ ]: good_results = results[698:895]
```

```
[ ]: center_of_rot = (725, 520, 50, 50)
```

```
[ ]: def pos_from_case(case):
         x,y,w,h = case
         return (x + (x+w))/2, (y + (y+h))/2
```

```
[ ]: pos_c = pos_from_case(center_of_rot)
     pos_r = [pos_from_case(i) for i in good_results]
```

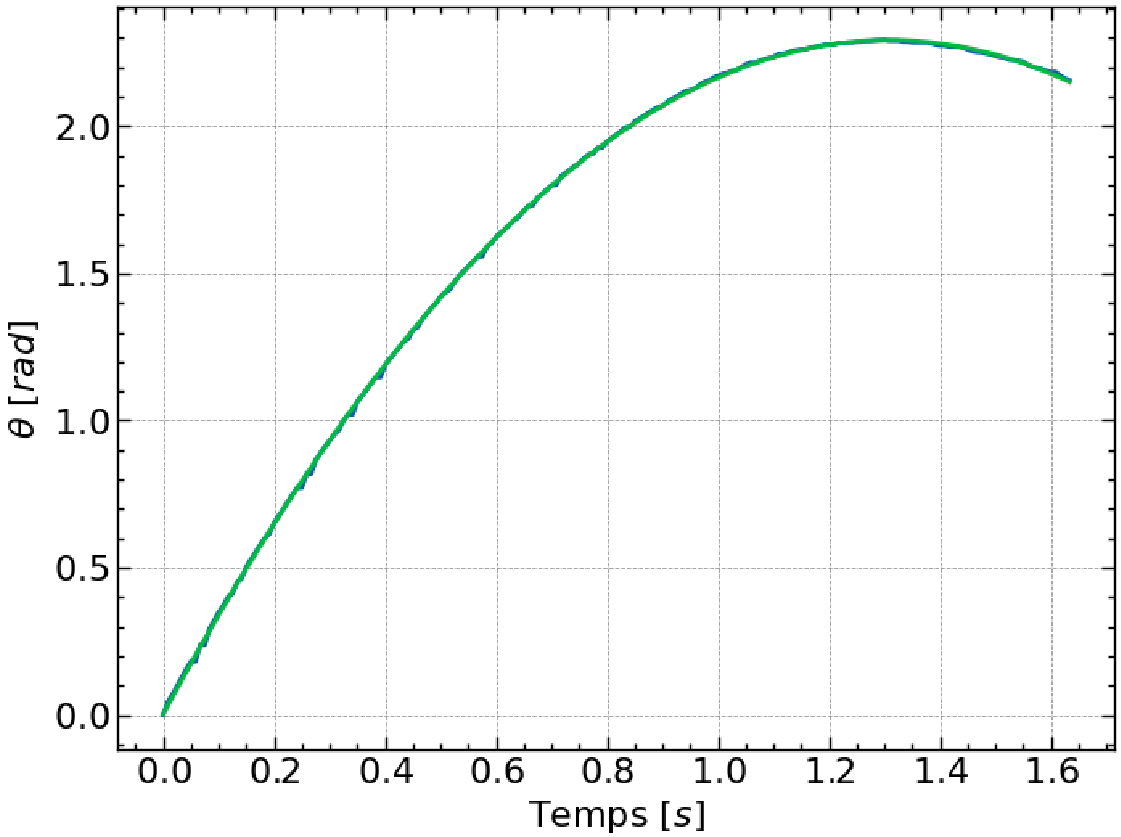
[ ]:
```
theta = []
cur_theta = 0

prev_dir = [pos_r[0][0]-pos_c[0], pos_r[0][1]-pos_c[1]]
for i in range(len(pos_r)):
    next_dir = [pos_r[i][0]-pos_c[0], pos_r[i][1]-pos_c[1]]
    cur_theta += (np.arctan2(next_dir[1], next_dir[0]) - np.
 ↪arctan2(prev_dir[1], prev_dir[0]))
    if np.abs(np.arctan2(next_dir[1], next_dir[0]) - np.arctan2(prev_dir[1],␣
 ↪prev_dir[0])) > np.pi:
        cur_theta -= 2*np.pi*np.sign(np.arctan2(next_dir[1], next_dir[0]) - np.
 ↪arctan2(prev_dir[1], prev_dir[0]))
    theta.append(cur_theta)
    prev_dir = next_dir

print(theta)
```

```
[0.0, 0.04380327424584418, 0.0698416768609268, 0.0992867396492505,
0.1275495712062367, 0.15595449000322364, 0.18201905017031114,
0.18201905017031114, 0.2389515692831007, 0.2389515692831007, 0.2927828500948424,
0.3211879679773309, 0.34958392252349124, 0.37242472153889855,
0.4007690803770129, 0.4123452161977905, 0.45257238524667115,
0.46425523408242664, 0.5045267572218198, 0.5292016199518361, 0.5549409253184747,
0.5792457025605646, 0.6020409778940783, 0.6165203331944827, 0.6535301930232286,
0.677958256811928, 0.7008074367585829, 0.7264356597051758, 0.7509199894769019,
0.7727436513269781, 0.7727436513269781, 0.81897851086037, 0.81897851086037,
0.8667841195764162, 0.8899530260598265, 0.9123015535090342, 0.9330765338721232,
0.9538260189455294, 0.9679485293908251, 1.0006258418614635, 1.0213182315819922,
1.0213182315819922, 1.0661535435207763, 1.0845792941324564, 1.1050027120437809,
1.127592081741108, 1.1480972337282869, 1.1480972337282869, 1.1893329524820495,
1.21019795998942, 1.2288326053319092, 1.2474552638681553, 1.2684219136458452,
1.2789162555822486, 1.307977051450964, 1.3168911630276963, 1.3443837882092968,
1.3638991060148522, 1.3834593268035202, 1.3990990771494554, 1.4226549287305748,
1.4357260796807565, 1.4449025663601027, 1.4710464392688025, 1.4921878553579853,
1.509053478089022, 1.5247949601652588, 1.5395340345996997, 1.5571171760895404,
1.5580645859313103, 1.591472126589244, 1.6027440275291782, 1.6249435734028412,
1.6391794890422515, 1.6513306885387768, 1.6669958368418631, 1.6819866731312596,
1.6935312223768102, 1.714364439907636, 1.7293941003665718, 1.7323076169659068,
1.759018221647629, 1.7707481960576934, 1.7857525693696894, 1.8004606004032104,
1.8004606004032104, 1.8301486172100225, 1.8419529392822762, 1.853769641384582,
1.8655954386913982, 1.8744687706282301, 1.8922196284374082, 1.9040521527878786,
1.9070096367713638, 1.927698993280915, 1.927698993280915, 1.9485366072897576,
1.9603508769518985, 1.9721444750595516, 1.9842028454698426, 1.9963044510043217,
1.9992525816152598, 2.0172848284758222, 2.026103159348146, 2.035339328217943,
2.0470652783572154, 2.0558307491893024, 2.065095592534302, 2.0685477417064915,
2.0836980975812613, 2.093032715595731, 2.101750882824084, 2.11043591008981,
2.119770917837757, 2.123350581258365, 2.1291213770028383, 2.141360053227973,
2.150731197394874, 2.1564557374427853, 2.165824690015021, 2.171346423114583,
2.1788986082757247, 2.1817433748751007, 2.1883002523291584, 2.191137412498376,
2.2014865823826257, 2.2099327891623206, 2.2165194937828696, 2.2165194937828696,
2.223117865758465, 2.2287268380080847, 2.2325303118036057, 2.2353276063504723,
2.2457710445948877, 2.2457710445948877, 2.2504804464235586, 2.2571166124294573,
2.2590485384822467, 2.2618255673915755, 2.2618255673915755, 2.2673614009753784,
2.270413002371429, 2.2731782554624105, 2.2778826592925525, 2.2778826592925525,
2.2778826592925525, 2.2833819829563646, 2.2833819829563646, 2.284535589685725,
2.284535589685725, 2.2872819809106284, 2.2872819809106284, 2.2900219916109803,
2.2900219916109803, 2.291976490458446, 2.293155105817337, 2.293934043189009,
2.291194324016465, 2.291194324016465, 2.291194324016465, 2.2872819809106284,
2.2872819809106284, 2.284535589685725, 2.284535589685725, 2.284535589685725,
2.284535589685725, 2.2825839692935235, 2.2778826592925525, 2.275123579106606,
2.275123579106606, 2.2712361133678387, 2.2712361133678387, 2.2712361133678387,
2.2673614009753784, 2.260734997269335, 2.255187965941525, 2.252405478834106,
2.2513405354124787, 2.2475062932250642, 2.2447252609039574, 2.239145801828297,
2.2353276063504737, 2.232530311803607, 2.225925130724845, 2.225016592038394,
2.223117865758467, 2.2193308094664204, 2.208053092128025, 2.2014556383691986,
2.1977053899746055, 2.1948711814898996, 2.18830025232916, 2.184583224013301,
2.184583224013301, 2.172353979448701, 2.1629708768132776, 2.157249388069803]
```

```
[ ]: time_1 = np.arange(len(theta))/frame_rate
     popt_1, _ = sp.optimize.curve_fit(quad, time_1, theta)
     popt_1
```

```
[ ]: array([-1.33778279e+00,  3.50021783e+00,  3.49643201e-03])
```

```
[ ]: plt.plot(time_1, theta)
     plt.plot(time_1, quad(time_1, *popt_1))
     plt.xlabel(r"Temps $[s]$")
     plt.ylabel(r"$\theta$ $[rad]$")
     plt.savefig("g70_t.png")
```

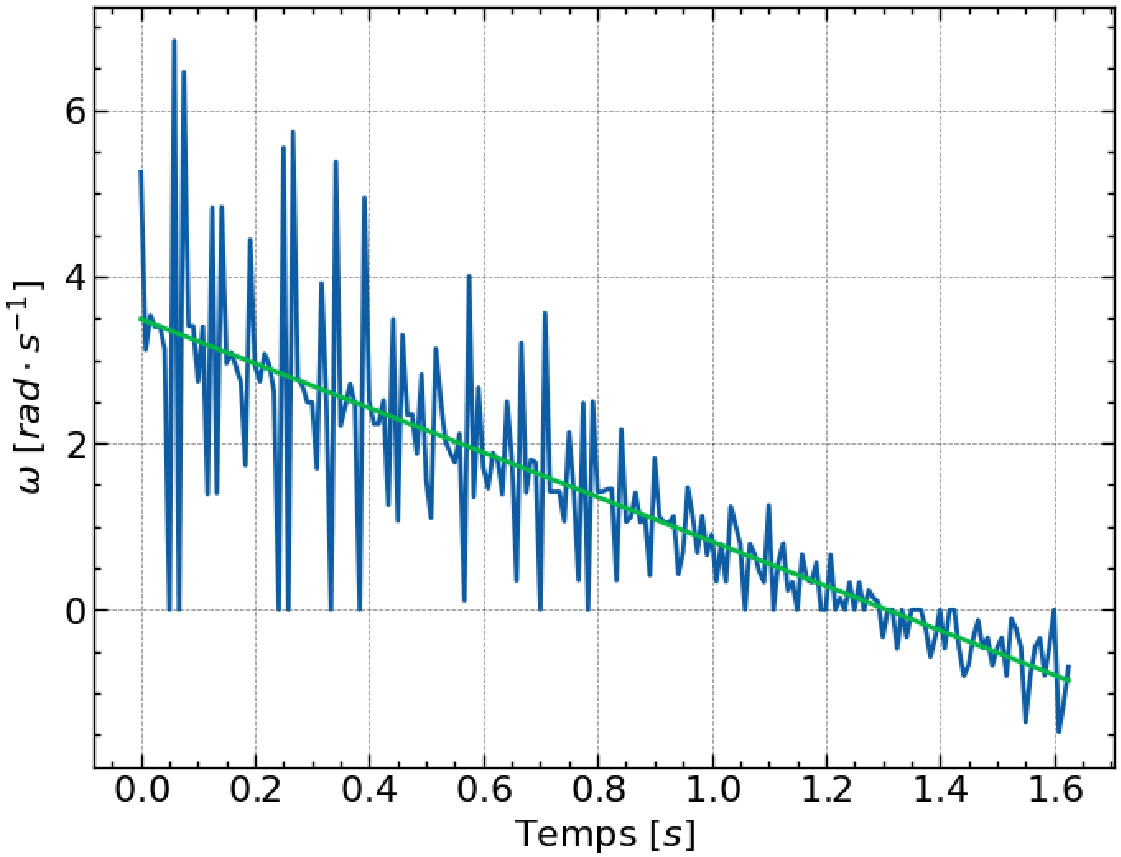


[ ]:
```
omega = []

for i in range(len(theta)-1):
    omega.append((theta[i+1]-theta[i])/(1/frame_rate))
```

[ ]:
```
time_2 = np.arange(len(omega))/frame_rate
popt_2, _ = sp.optimize.curve_fit(lin, time_2, omega)
popt_2
```

[ ]:
```
array([-2.67072506,  3.49072904])
```

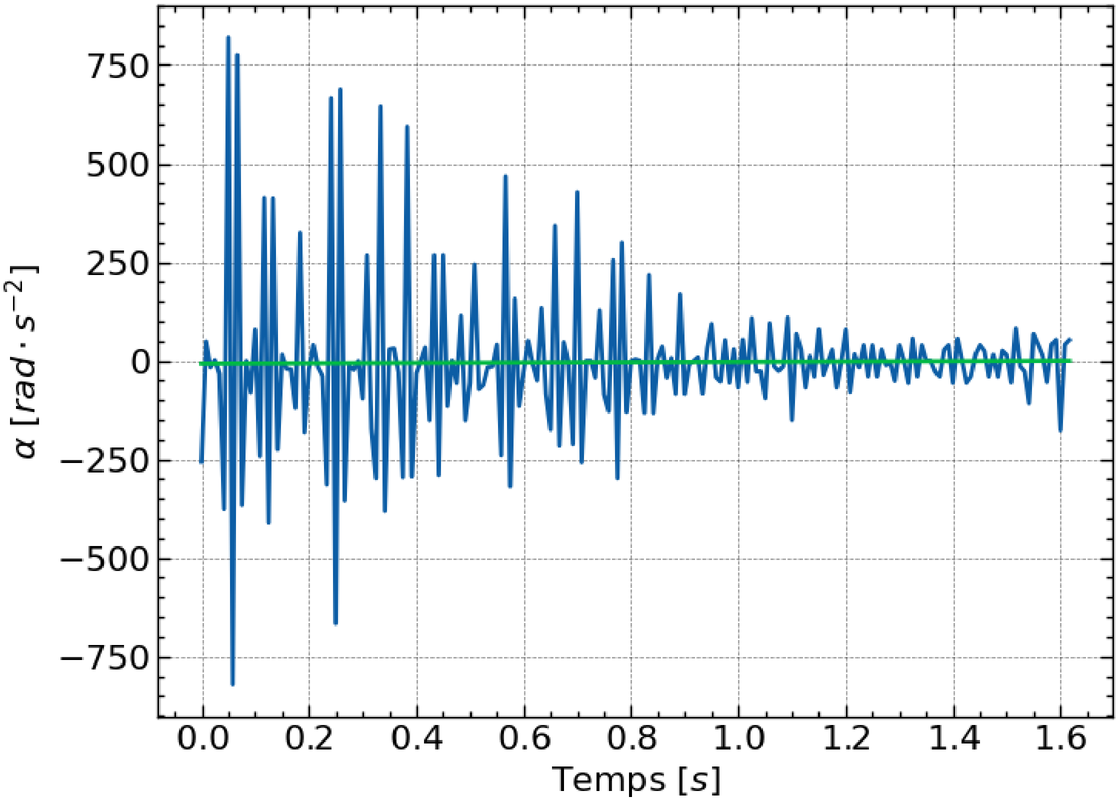
[ ]:
```
plt.plot(time_2, omega)
plt.plot(time_2, lin(time_2, *popt_2))
plt.xlabel(r"Temps $[s]$")
plt.ylabel(r"$\omega$ $[rad \cdot s^{-1}]$")
plt.savefig("g70_o.png")
```



[ ]:

```
alpha = []

for i in range(len(omega)-1):
    alpha.append((omega[i+1]-omega[i])/(1/frame_rate))

time_3 = np.arange(len(alpha))/frame_rate
popt_3, _ = sp.optimize.curve_fit(lin, time_3, alpha)
print(popt_3)

plt.plot(np.arange(len(alpha))/frame_rate, alpha)
plt.plot(np.arange(len(alpha))/frame_rate, lin(time_3, *popt_3))
plt.xlabel(r"Temps $[s]$")
plt.ylabel(r"$\alpha$ $[rad \cdot s^{-2}]$")
plt.savefig("g70_a.png")
```

```
[ 4.40400655 -7.21711223]
```



```
[ ]: acc_quad = popt_1[0]*2
     acc_lin = popt_2[0]
     print(acc_quad, acc_lin)
     acc = (acc_quad + acc_lin)/2
     Mom = 9.81 * m70 * r_g
     I_g70 = Mom/acc
     print(I_g70)
```

```
-2.675565570653598 -2.670725058482967
-0.011559977615728925
```

1.6 Vidéo m80

```
[ ]: video_path = "m80.mp4"
     cap = cv2.VideoCapture(video_path)
```

```
[ ]: frame_rate = cap.get(cv2.CAP_PROP_FPS)
     print("Frame rate: ", frame_rate)
```

```
Frame rate:  59.78746842677778
```

[ ]:
```
frames = []
while True:
    ret, frame = cap.read()
    if not ret:
        break
    frames.append(frame)
```

[ ]:
```
frame_rate = cap.get(cv2.CAP_PROP_FPS)
print("Frame rate: ", frame_rate)
```

```
Frame rate:  59.78746842677778
```

[ ]:
```
def disp_f(num):
    plt.imshow(frames[num])

widgets.interact(disp_f, num=widgets.IntSlider(min=0, max=len(frames)-1,␣
 ↪step=1))
```

```
interactive(children=(IntSlider(value=0, description='num', max=1152),␣
 ↪Output()), _dom_classes=('widget-intera…
```

[ ]:
```
<function __main__.disp_f(num)>
```

[ ]:
```
analyse = frames[678:957]
```

[ ]:
```
def pos_track(x,y):
    image = analyse[0].copy()
    w, h = 50, 50
    cv2.rectangle(image, (x, y), (x + w, y + h), (255, 0, 0), 4)
    plt.imshow(image)

widgets.interact(pos_track, x=widgets.IntSlider(min=0, max=len(analyse[0]),␣
 ↪step=1), y=widgets.IntSlider(min=0, max=len(analyse[0][0]), step=1))
```

```
interactive(children=(IntSlider(value=0, description='x', max=1080),␣
 ↪IntSlider(value=0, description='y', max=1…
```

[ ]:
```
<function __main__.pos_track(x, y)>
```

[ ]:
```
tracker = cv2.TrackerCSRT_create()

track_frames = analyse.copy()

init_trac_f = track_frames[0]
roi = (401, 294, 50, 50)

tracker.init(init_trac_f, roi)
```

```
results = []
frame_tracked = []
for i in range(1, len(track_frames)):
    # Read the next frame
    frame = track_frames[i]

    # Update the tracker
    success, roi = tracker.update(frame)

    if success:
        # Draw a bounding box around the tracked object
        (x, y, w, h) = tuple(map(int, roi))
        results += [(x,y,w,h)]
        cv2.rectangle(frame, (x, y), (x + w, y + h), (255, 0, 0), 4)
        frame_tracked.append(frame)
```

[ ]:

```
def disp_t(num):
    plt.imshow(frame_tracked[num])

widgets.interact(disp_t, num=widgets.IntSlider(min=0, max=len(frame_tracked)-1,␣
 ↪step=1))
```

```
interactive(children=(IntSlider(value=0, description='num', max=277), Output()),␣
 ↪_dom_classes=('widget-interac…
```

[ ]:

```
<function __main__.disp_t(num)>
```

[ ]:

```
good_results = results
```

[ ]:

```
center_of_rot = (705, 535, 50, 50)
```

[ ]:

```
def pos_from_case(case):
    x,y,w,h = case
    return (x + (x+w))/2, (y + (y+h))/2
```

[ ]:

```
pos_c = pos_from_case(center_of_rot)
pos_r = [pos_from_case(i) for i in good_results]
```

[ ]:

```
theta = []
cur_theta = 0

prev_dir = [pos_r[0][0]-pos_c[0], pos_r[0][1]-pos_c[1]]
for i in range(len(pos_r)):
    next_dir = [pos_r[i][0]-pos_c[0], pos_r[i][1]-pos_c[1]]
    cur_theta += (np.arctan2(next_dir[1], next_dir[0]) - np.
 ↪arctan2(prev_dir[1], prev_dir[0]))
```

```
    if np.abs(np.arctan2(next_dir[1], next_dir[0]) - np.arctan2(prev_dir[1],␣
 ↪prev_dir[0])) > np.pi:
        cur_theta -= 2*np.pi*np.sign(np.arctan2(next_dir[1], next_dir[0]) - np.
 ↪arctan2(prev_dir[1], prev_dir[0]))
    theta.append(cur_theta)
    prev_dir = next_dir

print(theta)
```

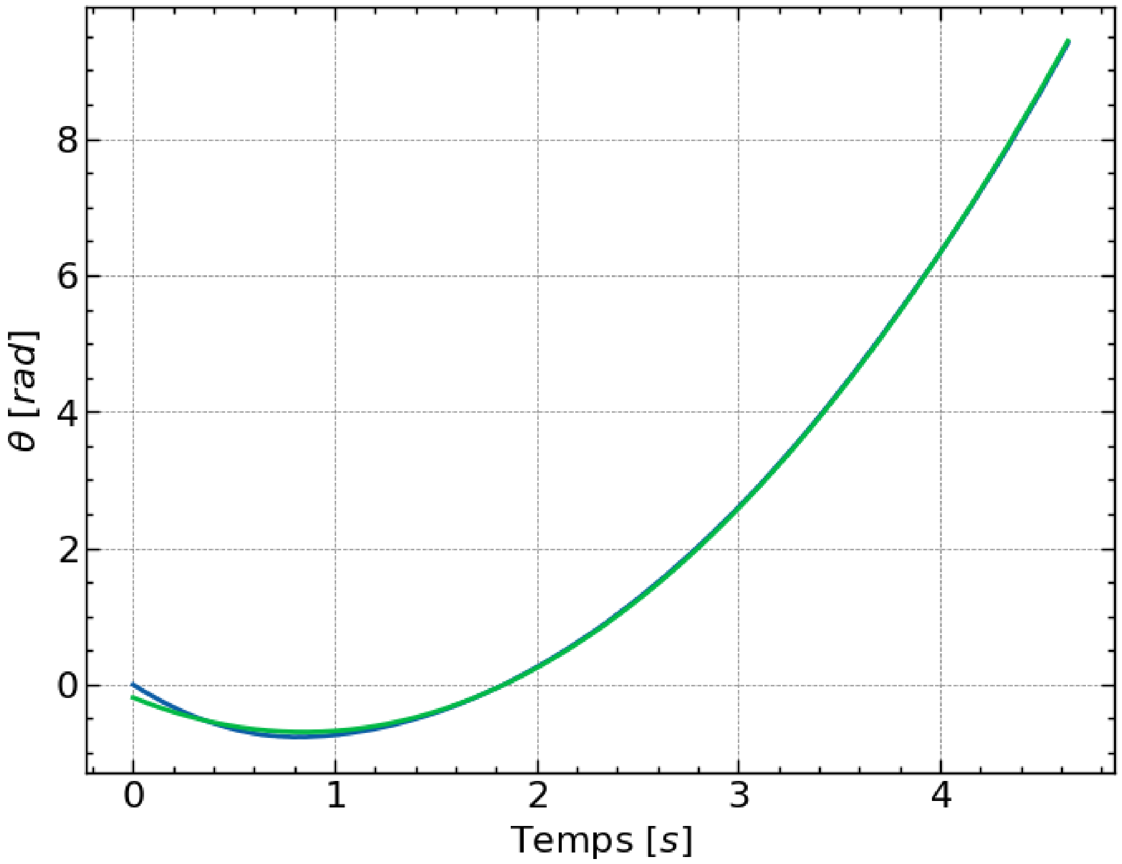
```
[0.0, -0.03351529685154597, -0.061766029641606934, -0.09542503022461757,
-0.12072140963968891, -0.14912421176569035, -0.1786368392925608,
-0.2063468739996459, -0.23483413978920975, -0.26032114218349056,
-0.2840963777029568, -0.3086339577490933, -0.3340273255975359,
-0.3569915570857578, -0.3781179566327788, -0.40132102303162354,
-0.42113884137910107, -0.4435170622073539, -0.4644126832071942,
-0.48426890882233753, -0.5029833012852252, -0.5221252777668921,
-0.5393300065382354, -0.5548421501731746, -0.5729908121751492,
-0.5847528264468131, -0.5992000290910728, -0.6173401912450336,
-0.6303070371103634, -0.6432870562466526, -0.6536892214798602,
-0.6692859648378877, -0.6784285556413399, -0.6900577068257081,
-0.6978368038801674, -0.7095828767580437, -0.7187494976922677,
-0.723929661199171, -0.7345226715542981, -0.73827896371031, -0.7446021725336851,
-0.7503308138486795, -0.7526248767176975, -0.7595392442860627,
-0.7606749101627779, -0.7647098921777196, -0.765837333220333,
-0.7698738123653417, -0.7709929890492679, -0.7687513642031663,
-0.7698738123653417, -0.7698738123653417, -0.7669615010141726,
-0.7618072900098274, -0.7621253927530556, -0.7606749101627779,
-0.7529191614956932, -0.7529191614956932, -0.7463064383017768,
-0.7437177368116359, -0.7371128130448916, -0.7345226715542981,
-0.7241496347987906, -0.7201497917183741, -0.7135650624385717,
-0.7057809991971964, -0.6992102671495459, -0.6914222204953893,
-0.6822724670518334, -0.6732271477673759, -0.6653540928977293,
-0.657565720759151, -0.6471779557737305, -0.6380855163021302,
-0.6250825898924859, -0.616013539094602, -0.6042897893621708,
-0.5939036396546249, -0.5809350074921786, -0.5665980570822482,
-0.5548421501731751, -0.5416589983274545, -0.5275825532089793,
-0.5132615708593313, -0.5000585957925896, -0.4831883958475043,
-0.46997286179655484, -0.45674754421391617, -0.43843164199807605,
-0.42367178447566767, -0.4054027152214963, -0.39157438894747143,
-0.3717696009014113, -0.3520198284422409, -0.33480455528665765,
-0.3184563691674538, -0.29887735591835707, -0.28076681946779747,
-0.2586390861644001, -0.2392366312709706, -0.21788547722609808,
-0.19885556506965996, -0.17689169871138155, -0.15611033268872365,
-0.13529792907779692, -0.11270667540387747, -0.08832379172541227,
-0.06533588503583321, -0.04459526777793954, -0.018537975729311373,
0.0036005351134389407, 0.02932859282125211, 0.056671195990560363,
0.08234614141125851, 0.10414238867039272, 0.12967977180296186,
0.1569209158939393, 0.18599094483456513, 0.21330575985119093,
0.2406194928310046, 0.2663077346392293, 0.29714455513679905,
0.32489141549314704, 0.352666812223279, 0.3826173836282978, 0.41258298116626335,
0.439560734688611, 0.47178809226504104, 0.5045701167486267, 0.5349416162169713,
0.5682873845041867, 0.6007315325322446, 0.6324386763190064, 0.6642548257839846,
0.6967149729404729, 0.7279573886552366, 0.762247398048965, 0.7960563483786403,
0.8323805828912116, 0.8682686616042086, 0.9005718717660649, 0.9391223504812416,
0.9738559549086832, 1.0155176288692986, 1.0488954166925235, 1.0897593733189992,
1.1232476753906522, 1.1607262047963893, 1.2017357674873899, 1.2425404598056449,
1.2784204755993396, 1.3193489318148541, 1.3569702229906586, 1.400269332093807,
1.4435189036241018, 1.4842516427012626, 1.5255083238558855, 1.5679186368966747,
1.6049387527706047, 1.6526764695047405, 1.6947937930691213, 1.7406136106912418,
1.7827678091293968, 1.8252377356380227, 1.867784507686145, 1.9124062318740256,
1.9600039236206983, 2.0044091316661583, 2.051922666410493, 2.1007244267572323,
2.1443471007639427, 2.190324294230517, 2.2389220490937696, 2.284946251802574,
2.3323552406875177, 2.38205340930373, 2.4318270987734163, 2.4813220564016016,
2.5309194814717464, 2.580579918829604, 2.6359678873518373, 2.683698056245706,
2.734495736874641, 2.78840570400252, 2.838017583303894, 2.893381973358346,
2.945950662465613, 2.9986295013085016, 3.0520599305802047, 3.1072552157173576,
3.1632499167543067, 3.218939797348542, 3.2728185583222453, 3.3283985941689336,
3.385801081882029, 3.4439177431037358, 3.5024532443554404, 3.5577581110124736,
3.6140447630958468, 3.6732729769146797, 3.733057671478127, 3.790204615740236,
3.8524872776781613, 3.9125896180263444, 3.9733344222073557, 4.033756222259003,
4.093942422209226, 4.156795913792738, 4.219698911218707, 4.283731177842544,
4.345207434504035, 4.410415992118265, 4.472411708258035, 4.535483823085386,
4.603141310063503, 4.666806918459837, 4.729016564883576, 4.799429105628047,
4.862077692120075, 4.928397400197243, 4.998736607348157, 5.060861877640604,
5.128854943193762, 5.197591884568309, 5.262810165896122, 5.33191694782031,
5.400079390942926, 5.468826179014435, 5.5397380123988675, 5.610133369831394,
5.6800529402848445, 5.750167258067182, 5.818321472894314, 5.889405417110121,
5.962276645158545, 6.03342260854437, 6.100526705743979, 6.176700246451693,
6.246086747628377, 6.321556622524583, 6.39466584256261, 6.463729287219111,
6.538055147935601, 6.611977409768528, 6.686059471452921, 6.761848433393836,
6.836715270182072, 6.913167578962673, 6.990492810541079, 7.065769535858834,
7.141367826906452, 7.223604493578726, 7.298702936048884, 7.379108408332678,
7.458634247536725, 7.536346282684952, 7.614058317833178, 7.695582639847269,
7.780331897478861, 7.860171522901564, 7.939517270496469, 8.02196837598242,
8.10862189486569, 8.190190151871837, 8.275266314731395, 8.356614027637187,
8.440619253206581, 8.525545775096333, 8.611473549028423, 8.6956726801845,
8.778873242599582, 8.872978665907704, 8.95976200232492, 9.048435313963582,
9.13449896728733, 9.22154513149358, 9.312807804065923, 9.39986652927939]
```

```
[ ]: time_1 = np.arange(len(theta))/frame_rate
     popt_1, _ = sp.optimize.curve_fit(quad, time_1, theta)
     popt_1
```

```
[ ]: array([ 0.70658563, -1.19618068, -0.1905181 ])
```

[ ]:

```
plt.plot(time_1, theta)
plt.plot(time_1, quad(time_1, *popt_1))
plt.xlabel(r"Temps $[s]$")
plt.ylabel(r"$\theta$ $[rad]$")
plt.savefig("m80_t.png")
```



[ ]:

```
omega = []

for i in range(len(theta)-1):
    omega.append((theta[i+1]-theta[i])/(1/frame_rate))
```

[ ]:

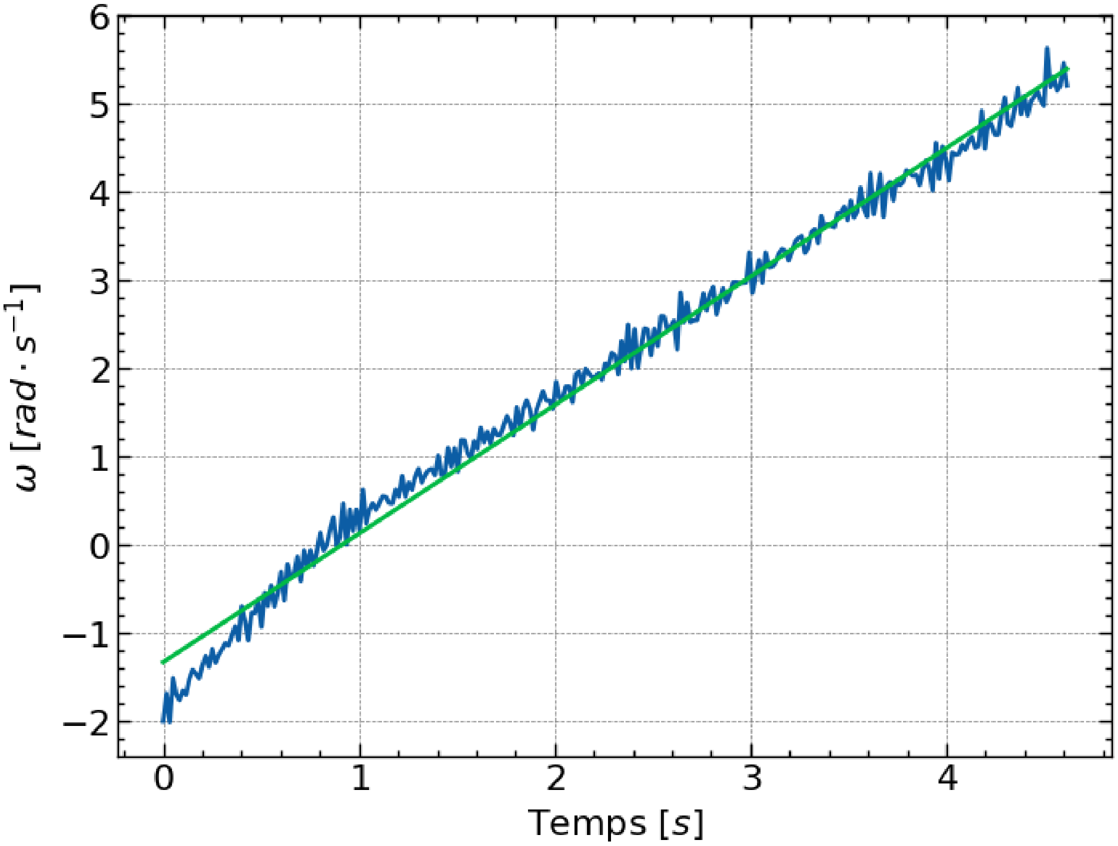
```
time_2 = np.arange(len(omega))/frame_rate
popt_2, _ = sp.optimize.curve_fit(lin, time_2, omega)
popt_2
```

[ ]:

```
array([ 1.45686417, -1.33383892])
```

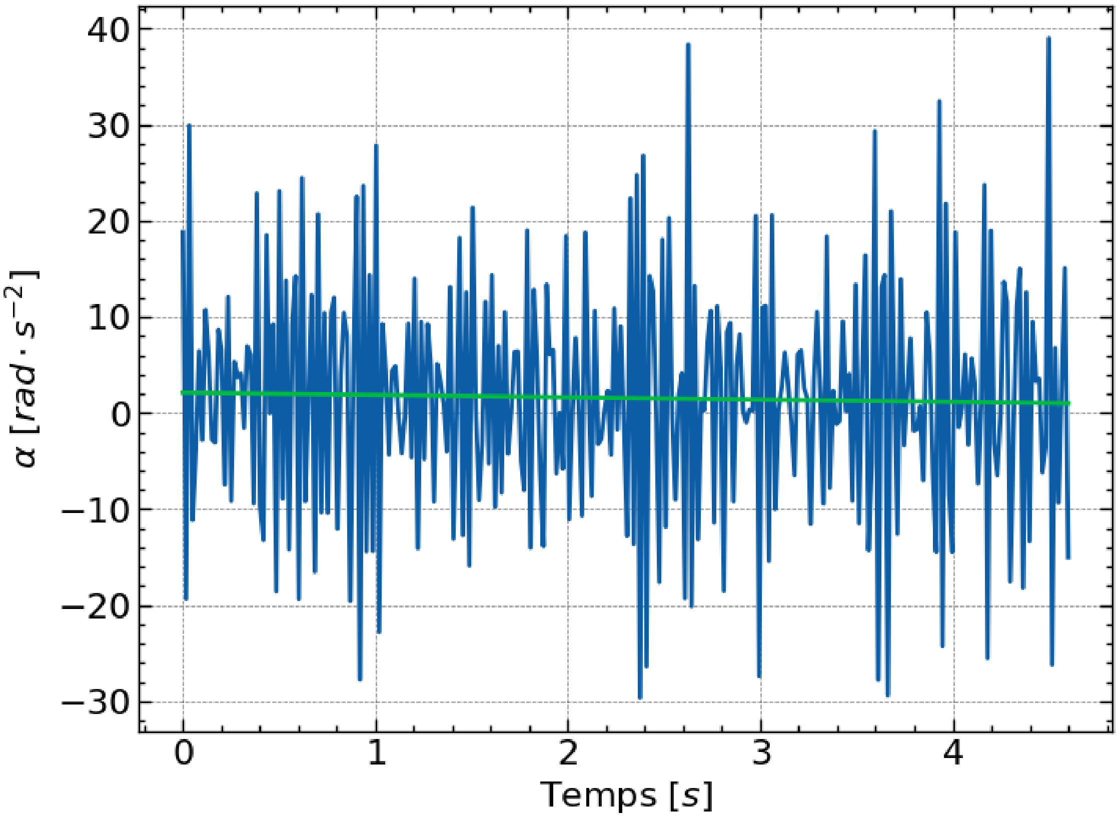
[ ]:

```
plt.plot(time_2, omega)
plt.plot(time_2, lin(time_2, *popt_2))
plt.xlabel(r"Temps $[s]$")
plt.ylabel(r"$\omega$ $[rad \cdot s^{-1}]$")
plt.savefig("m80_o.png")
```



[ ]:
```
alpha = []

for i in range(len(omega)-1):
    alpha.append((omega[i+1]-omega[i])/(1/frame_rate))

time_3 = np.arange(len(alpha))/frame_rate
popt_3, _ = sp.optimize.curve_fit(lin, time_3, alpha)
print(popt_3)

plt.plot(np.arange(len(alpha))/frame_rate, alpha)
plt.plot(np.arange(len(alpha))/frame_rate, lin(time_3, *popt_3))
plt.xlabel(r"Temps $[s]$")
plt.ylabel(r"$\alpha$ $[rad \cdot s^{-2}]$")
plt.savefig("m80_a.png")
```

```
[-0.24202158  2.11818713]
```



```
[ ]: acc_quad = popt_1[0]*2
     acc_lin = popt_2[0]
     print(acc_quad, acc_lin)
     acc = (acc_quad + acc_lin)/2
     Mom = 9.81 * m80 * r_m
     I_m80 = Mom/acc
     print(I_m80)
```

```
1.4131712550841267 1.4568641739241976
0.016406766106113954
```

1.7 Vidéo m100

```
[ ]: video_path = "m100.mp4"
     cap = cv2.VideoCapture(video_path)
```

```
[ ]: frame_rate = cap.get(cv2.CAP_PROP_FPS)
     print("Frame rate: ", frame_rate)
```

```
Frame rate:  59.78130509097191
```

[ ]:

```
frames = []
while True:
    ret, frame = cap.read()
    if not ret:
        break
    frames.append(frame)
```

[ ]:

```
frame_rate = cap.get(cv2.CAP_PROP_FPS)
print("Frame rate: ", frame_rate)
```

```
Frame rate:  59.78130509097191
```

[ ]:

```
def disp_f(num):
    plt.imshow(frames[num])

widgets.interact(disp_f, num=widgets.IntSlider(min=0, max=len(frames)-1,␣
 ↪step=1))
```

```
interactive(children=(IntSlider(value=0, description='num', max=791), Output()),␣
 ↪_dom_classes=('widget-interac…
```

[ ]:

```
<function __main__.disp_f(num)>
```

[ ]:

```
analyse = frames[356:662]
```

[ ]:

```
def pos_track(x,y):
    image = analyse[0].copy()
    w, h = 50, 50
    cv2.rectangle(image, (x, y), (x + w, y + h), (255, 0, 0), 4)
    plt.imshow(image)

widgets.interact(pos_track, x=widgets.IntSlider(min=0, max=len(analyse[0]),␣
 ↪step=1), y=widgets.IntSlider(min=0, max=len(analyse[0][0]), step=1))
```

```
interactive(children=(IntSlider(value=0, description='x', max=1080),␣
 ↪IntSlider(value=0, description='y', max=1…
```

[ ]:

```
<function __main__.pos_track(x, y)>
```

[ ]:

```
tracker = cv2.TrackerCSRT_create()

track_frames = analyse.copy()

init_trac_f = track_frames[0]
roi = (418, 279, 50, 50)

tracker.init(init_trac_f, roi)
```

```
results = []
frame_tracked = []
for i in range(1, len(track_frames)):
    # Read the next frame
    frame = track_frames[i]

    # Update the tracker
    success, roi = tracker.update(frame)

    if success:
        # Draw a bounding box around the tracked object
        (x, y, w, h) = tuple(map(int, roi))
        results += [(x,y,w,h)]
        cv2.rectangle(frame, (x, y), (x + w, y + h), (255, 0, 0), 4)
        frame_tracked.append(frame)
```

[ ]:

```
def disp_t(num):
    plt.imshow(frame_tracked[num])

widgets.interact(disp_t, num=widgets.IntSlider(min=0, max=len(frame_tracked)-1,␣
 ↪step=1))
```

```
interactive(children=(IntSlider(value=0, description='num', max=304), Output()),␣
 ↪_dom_classes=('widget-interac…
```

[ ]: `<function __main__.disp_t(num)>`

[ ]:

```
good_results = results[0:277]
```

[ ]:

```
center_of_rot = (705, 535, 50, 50)
```

[ ]:

```
def pos_from_case(case):
    x,y,w,h = case
    return (x + (x+w))/2, (y + (y+h))/2
```

[ ]:

```
pos_c = pos_from_case(center_of_rot)
pos_r = [pos_from_case(i) for i in good_results]
```

[ ]:

```
theta = []
cur_theta = 0

prev_dir = [pos_r[0][0]-pos_c[0], pos_r[0][1]-pos_c[1]]
for i in range(len(pos_r)):
    next_dir = [pos_r[i][0]-pos_c[0], pos_r[i][1]-pos_c[1]]
    cur_theta += (np.arctan2(next_dir[1], next_dir[0]) - np.
 ↪arctan2(prev_dir[1], prev_dir[0]))
```

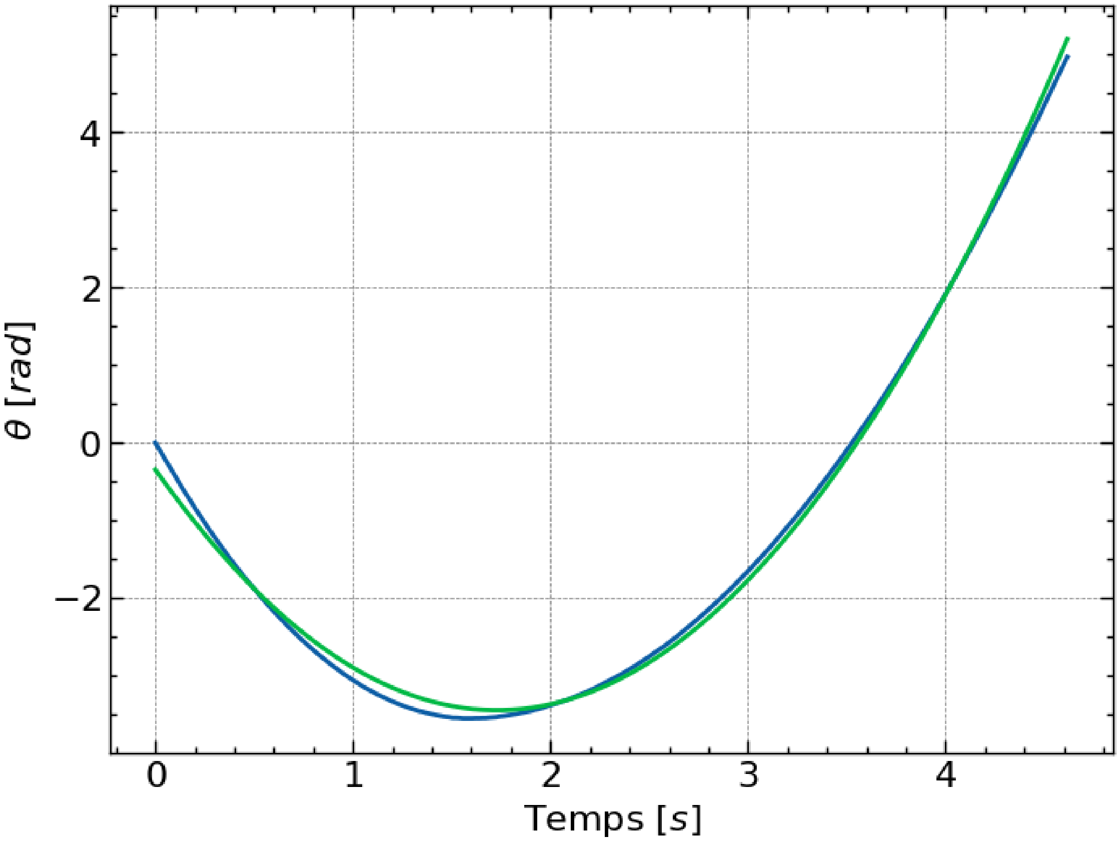
```
    if np.abs(np.arctan2(next_dir[1], next_dir[0]) - np.arctan2(prev_dir[1],␣
 ↪prev_dir[0])) > np.pi:
        cur_theta -= 2*np.pi*np.sign(np.arctan2(next_dir[1], next_dir[0]) - np.
 ↪arctan2(prev_dir[1], prev_dir[0]))
    theta.append(cur_theta)
    prev_dir = next_dir

print(theta)
```

```
[0.0, -0.07118707274296376, -0.14414895625611068, -0.21949067471002426,
-0.28641815152193795, -0.35986876501066956, -0.43131650120932497,
-0.5034937685398377, -0.575541813344806, -0.6408766570481141,
-0.7060875973123704, -0.7740333815239029, -0.8450845470731321,
-0.9069616148433948, -0.9696355473074374, -1.0338017887510467,
-1.1007296210767783, -1.1623734616490187, -1.2212177431155604,
-1.2822995085797584, -1.3402878613562117, -1.4028677249995365,
-1.4615731774941763, -1.514962208937496, -1.5664169306634075,
-1.625407819080091, -1.6821973226303437, -1.7367742529236518,
-1.7877313730111988, -1.8411654063092386, -1.889558629680698,
-1.9412060628221415, -1.9921242364339742, -2.0398838107291057,
-2.0896915650714023, -2.1363644899388117, -2.180564378730214,
-2.227277586070473, -2.272633371539565, -2.3153153762374794,
-2.3621337704532603, -2.4042180450215325, -2.444361315428644,
-2.4864381972737144, -2.529011990199563, -2.5673941131072477,
-2.607444232533383, -2.644421986677937, -2.6825742920971734, -2.717821516037558,
-2.75149222183164, -2.7866330285017034, -2.820457241879943, -2.8560139376152027,
-2.8857654457631874, -2.9168965116969594, -2.9500140226305462,
-2.9811423618890394, -3.010378283091321, -3.039561535024689, -3.067094645053573,
-3.0925905403542004, -3.1220126297915645, -3.147833997312608,
-3.172157822686961, -3.194408647588779, -3.2174062094453624,
-3.2418756020298476, -3.2627879288633332, -3.2837075764451935,
-3.3058241195851257, -3.323182231353102, -3.3416648595177367,
-3.3589757434302783, -3.3769301399739384, -3.393222603397167,
-3.408835518180258, -3.422757019572278, -3.4366734987886978, -3.449716702914652,
-3.4636415500480773, -3.4710125228910815, -3.4841623870289204,
-3.493997560608198, -3.5038116895957816, -3.5120023896170376,
-3.518591786675492, -3.5283652122957667, -3.53337426869702, -3.538396544142248,
-3.5465820380716617, -3.5465820380716617, -3.549101705047285,
-3.5541503945682327, -3.5541503945682327, -3.554757067901286,
-3.5541503945682327, -3.5516244993282178, -3.5522371659676, -3.549720342083902,
-3.549720342083902, -3.544065524523467, -3.5415521902997185, -3.538396544142248,
-3.530201664864766, -3.525865703331547, -3.5201872602190143, -3.513788102004675,
-3.506300384414726, -3.498845254144282, -3.4906609130780906,
-3.4824736085929264, -3.4742844371501733, -3.4660944967126803,
-3.45545895267222, -3.4424136573421533, -3.4333490073707975, -3.421825889958804,
-3.4078707460720805, -3.3939163518405597, -3.382345920549728,
-3.368393525185565, -3.353361212532823, -3.3394147678869293, -3.325495691588395,
-3.3104145681336723, -3.293024432203625, -3.275654618679972, -3.258315548617761,
-3.2410175323613766, -3.2223828276936874, -3.2022994450745816,
-3.183610032627676, -3.1649336672717983, -3.1426580324077724,
-3.1224408036039772, -3.100202572816544, -3.0783012111263126, -3.05614105262812,
-3.032319318098347, -3.0085668725049355, -2.9847969061366797,
-2.960976118147811, -2.933466025343005, -2.9096051976328576, -2.883729252513021,
-2.8542033294181506, -2.826197737821726, -2.7981628030673305,
-2.7715008403971044, -2.742180855751469, -2.709767415747735, -2.680838520147774,
-2.651913497397791, -2.624057487724216, -2.5903970106399155, -2.555201229493754,
-2.5248643019158905, -2.492814334931964, -2.457570604481803, -2.420459985515351,
-2.386360882227043, -2.3518862262480513, -2.3183987798513908,
-2.279175064386302, -2.2428175602093905, -2.2039629320977374,
-2.1676043233690807, -2.1286110377301424, -2.0919301264992542,
-2.049654214611152, -2.007869008835491, -1.9676002173473524, -1.929790219121801,
-1.889558629680699, -1.8443081751490797, -1.8023362441774506,
-1.759342419283589, -1.7139246669530488, -1.6719434858872564,
-1.6264562973936276, -1.5796390726280922, -1.5354230529820345,
-1.49113589420057, -1.4451772169599124, -1.395452789332659, -1.3494396101703021,
-1.2994906664164954, -1.2489857504474426, -1.2012196630177652,
-1.152009487710267, -1.1007296210767792, -1.050520054470486, -1.000154189035909,
-0.9481680192221802, -0.8936880581592215, -0.8430343401725966,
-0.7872891009503546, -0.7348828896653856, -0.6826389277389038,
-0.6252150243203634, -0.5703362241063727, -0.5127119769949733,
-0.46000038214890093, -0.4005621747020305, -0.3442695597874965,
-0.2890232535497277, -0.23181132462246357, -0.17224197555865084,
-0.11142468073158573, -0.05058421605487107, 0.01096140505501575,
0.0703655225368558, 0.13116396269941832, 0.1937171041358554,
0.25304393357196586, 0.3181897131251281, 0.38166743149195526,
0.44099923068603086, 0.509676490165152, 0.573983336310905, 0.6391278082697731,
0.703701115888465, 0.7706243116709621, 0.8361373253173638, 0.9065037776774618,
0.974123246571755, 1.0448264188294218, 1.1133538655861708, 1.1828570805960434,
1.25125697078665, 1.3238975912006683, 1.3927274337686786, 1.463846315108028,
1.5353731636533716, 1.6098299856184919, 1.6798678092784454, 1.7571679328839451,
1.827499244227829, 1.903439733816907, 1.9756738752657692, 2.0548412278125285,
2.1292508824405694, 2.208152820034724, 2.2832434126832233, 2.362733093857562,
2.4383918023167044, 2.518851445768992, 2.594361162111428, 2.6729689385175552,
2.7523172315408764, 2.833468604264933, 2.91479178199402, 2.9982216440734115,
3.077293923529852, 3.157106789243204, 3.239768364736293, 3.3241208478360007,
3.4087740980108383, 3.4900129171841714, 3.570970393242713, 3.6567497584107347,
3.7402950454817843, 3.8256147026977834, 3.910044258459793, 3.996249501431392,
4.081812312795266, 4.170323066612406, 4.257001847974473, 4.348410023875045,
4.4355114812512095, 4.5273271943531395, 4.615674703922185, 4.70874237949759,
4.799394911025825, 4.885961383117804, 4.973203543208877]
```

[ ]:
```
time_1 = np.arange(len(theta))/frame_rate
popt_1, _ = sp.optimize.curve_fit(quad, time_1, theta)
popt_1
```

```
[ ]: array([ 1.03777155, -3.58824855, -0.3479185 ])
```

```
[ ]: plt.plot(time_1, theta)
     plt.plot(time_1, quad(time_1, *popt_1))
     plt.xlabel(r"Temps $[s]$")
     plt.ylabel(r"$\theta$ $[rad]$")
     plt.savefig("m100_t.png")
```

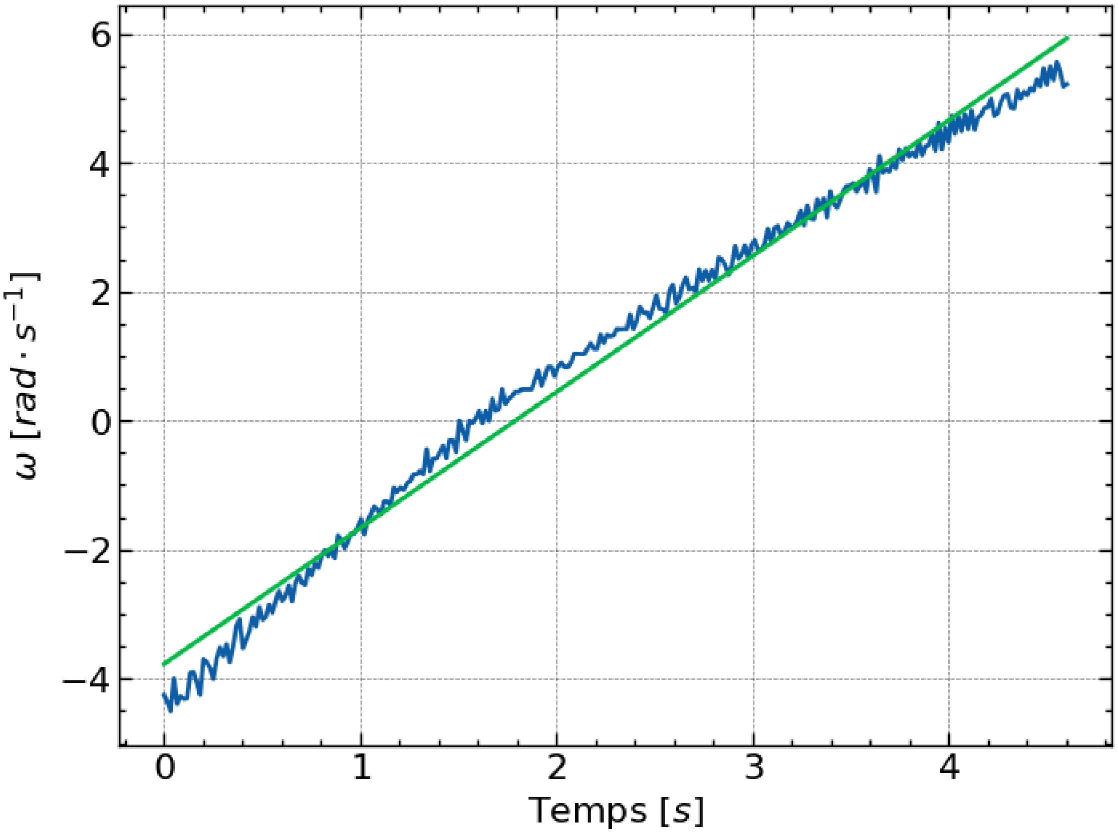


```
[ ]: omega = []

     for i in range(len(theta)-1):
         omega.append((theta[i+1]-theta[i])/(1/frame_rate))
```

```
[ ]: time_2 = np.arange(len(omega))/frame_rate
     popt_2, _ = sp.optimize.curve_fit(lin, time_2, omega)
     popt_2
```

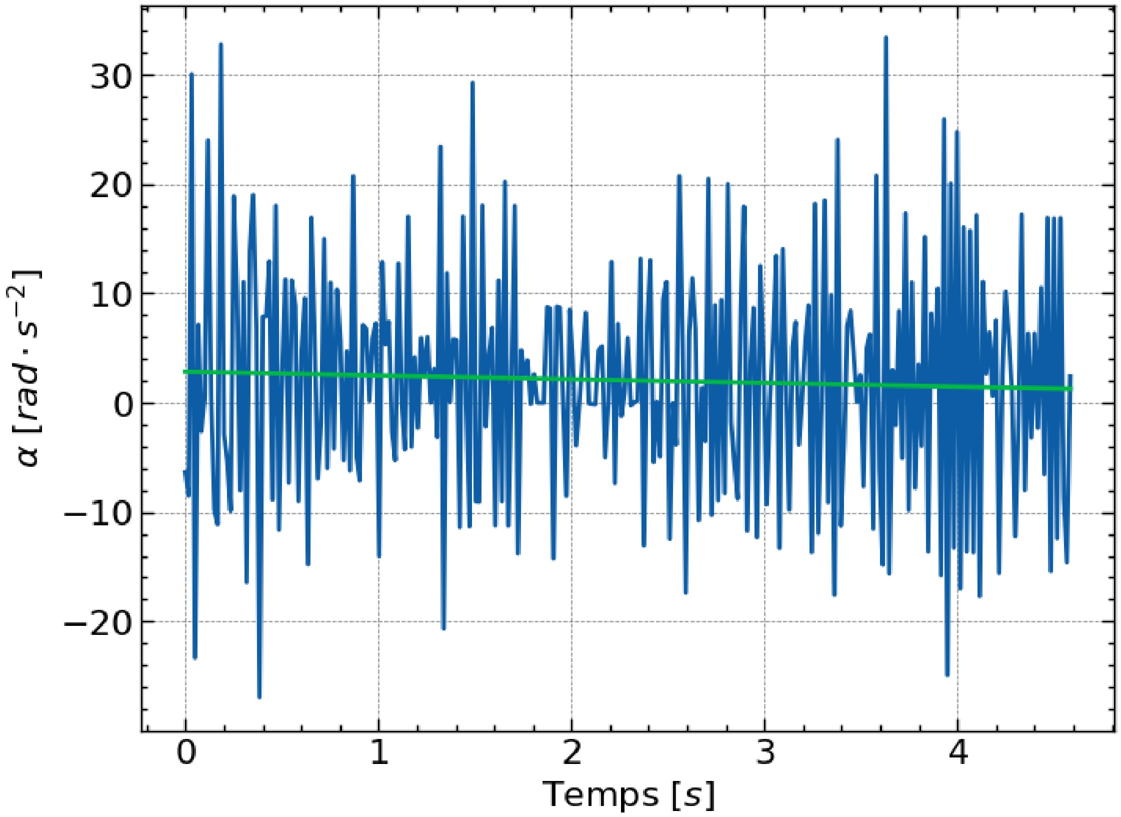
```
[ ]: array([ 2.10807802, -3.7714946 ])
```

[ ]:
```
plt.plot(time_2, omega)
plt.plot(time_2, lin(time_2, *popt_2))
plt.xlabel(r"Temps $[s]$")
plt.ylabel(r"$\omega$ $[rad \cdot s^{-1}]$")
plt.savefig("m100_o.png")
```



[ ]:
```
alpha = []

for i in range(len(omega)-1):
    alpha.append((omega[i+1]-omega[i])/(1/frame_rate))

time_3 = np.arange(len(alpha))/frame_rate
popt_3, _ = sp.optimize.curve_fit(lin, time_3, alpha)
print(popt_3)

plt.plot(np.arange(len(alpha))/frame_rate, alpha)
plt.plot(np.arange(len(alpha))/frame_rate, lin(time_3, *popt_3))
plt.xlabel(r"Temps $[s]$")
plt.ylabel(r"$\alpha$ $[rad \cdot s^{-2}]$")
plt.savefig("m100_a.png")
```

```
[-0.33995123  2.83795285]
```



[ ]:
```
acc_quad = popt_1[0]*2
acc_lin = popt_2[0]
print(acc_quad, acc_lin)
acc = (acc_quad + acc_lin)/2
Mom = 9.81 * m100 * r_m
I_m100 = Mom/acc
print(I_m100)
```

```
2.0755430982955625 2.1080780173316844
0.014069151668667558
```

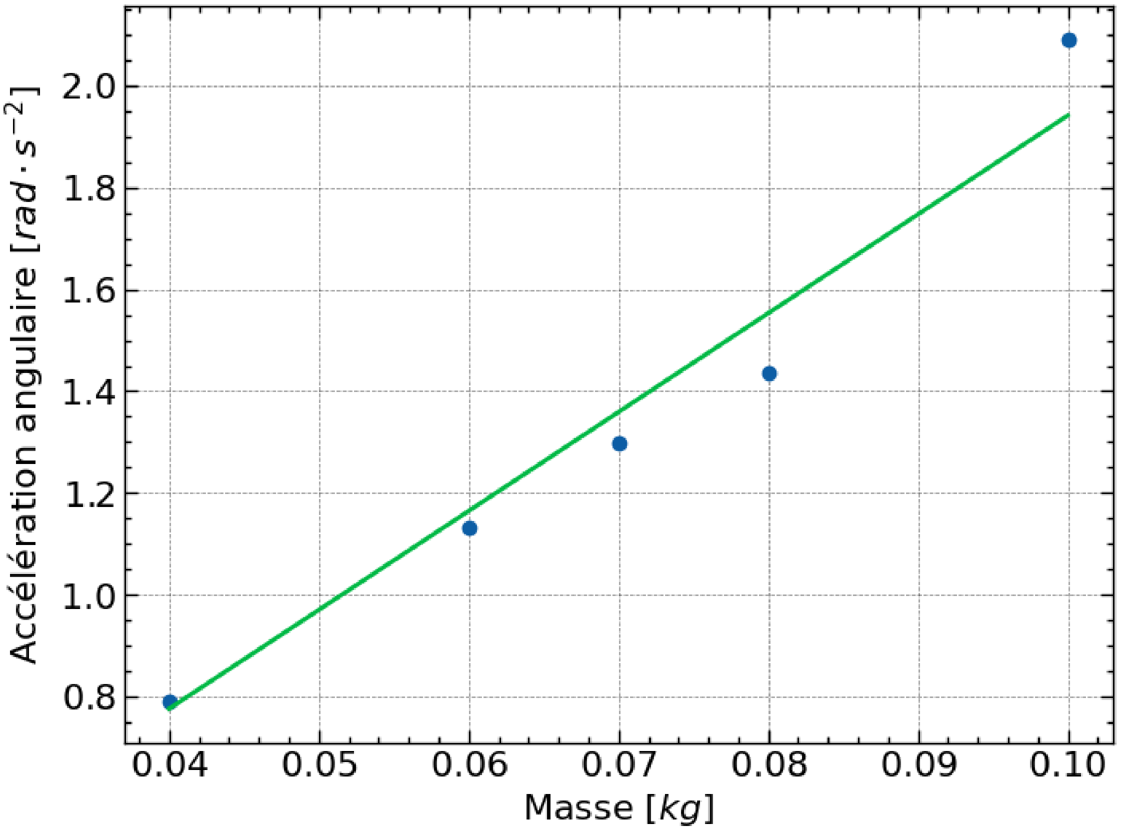
2 Analyse résultats

[ ]:
```
tabular = pd.DataFrame({
              "Mass": [40e-3, 60e-3, 70e-3, 70e-3, 70e-3, 80e-3, 100e-3],
              "Radius": [r_m, r_m, r_p, r_m, r_g, r_m, r_m],
              "Moment of Inertia": [I_m40, I_m60, I_p70, I_m70, np.abs(I_g70),␣
 ↪I_m80, I_m100]
              })
```

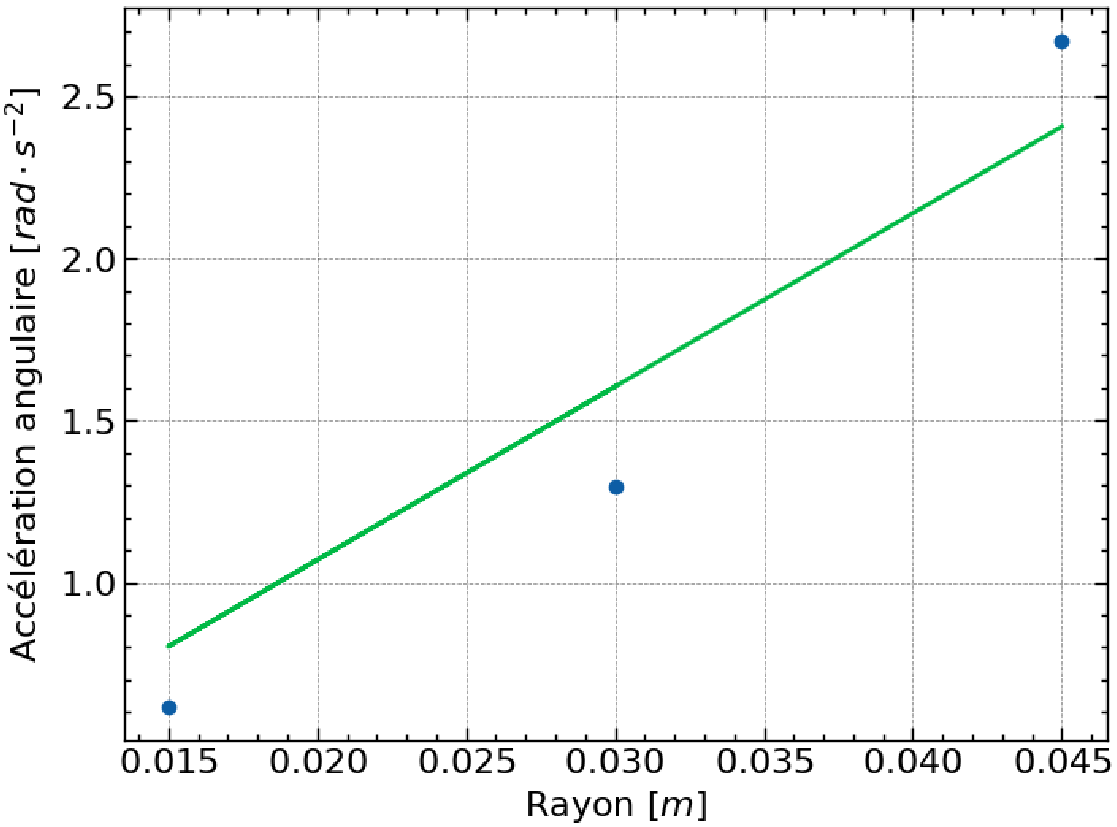
```
tabular["Angular Acceleration"] = tabular["Mass"]*9.81*tabular["Radius"]/
 ↪tabular["Moment of Inertia"]

tabular
```

```
[ ]:    Mass  Radius  Moment of Inertia  Angular Acceleration
     0  0.04   0.030           0.014886              0.790816
     1  0.06   0.030           0.015582              1.133199
     2  0.07   0.015           0.016659              0.618307
     3  0.07   0.030           0.015878              1.297437
     4  0.07   0.045           0.011560              2.673145
     5  0.08   0.030           0.016407              1.435018
     6  0.10   0.030           0.014069              2.091811
```

```
[ ]: # Plot a graph of the acceleration against the mass for all the measure with␣
      ↪radius r_m
     plt.plot(tabular["Mass"][tabular["Radius"] == r_m], tabular["Angular␣
      ↪Acceleration"][tabular["Radius"] == r_m], "o")
     popt_fr, _ = sp.optimize.curve_fit(prop:=lambda x,a: a*x,␣
      ↪tabular["Mass"][tabular["Radius"] == r_m], tabular["Angular␣
      ↪Acceleration"][tabular["Radius"] == r_m])
     plt.plot(np.array([40e-3, 100e-3]), prop(np.array([40e-3, 100e-3]), *popt_fr))
     plt.xlabel(r"Masse $[kg]$")
     plt.ylabel(r"Accélération angulaire $[rad \cdot s^{-2}]$")
     plt.savefig("prop_masse.png")
```



```
[ ]: # Plot a graph of the acceleration against the radius for all the measure with␣
      ↪mass 70g
     plt.plot(tabular["Radius"][tabular["Mass"] == 70e-3], tabular["Angular␣
      ↪Acceleration"][tabular["Mass"] == 70e-3], "o")
     popt_fr, _ = sp.optimize.curve_fit(prop:=lambda x,a: a*x,␣
      ↪tabular["Radius"][tabular["Mass"] == 70e-3], tabular["Angular␣
      ↪Acceleration"][tabular["Mass"] == 70e-3])
     plt.plot(np.array([r_m, r_p, r_g]), prop(np.array([r_m, r_p, r_g]), *popt_fr))
     plt.xlabel(r"Rayon $[m]$")
     plt.ylabel(r"Accélération angulaire $[rad \cdot s^{-2}]$")
     plt.savefig("prop_rayon.png")
```



[ ]:

```
# average and std for the moment of inertia
I_mean = np.mean(tabular["Moment of Inertia"])
I_std = np.std(tabular["Moment of Inertia"])
print(I_mean, I_std)
rel = I_std/I_mean
print(f"Relative uncertainty: {rel*100}%")
```

```
0.015005949828400644 0.0016281219932735526
Relative uncertainty: 10.849842974898712%
```

[ ]:

```
tab_report = pd.DataFrame()
tab_report["M [kg]"] = tabular["Mass"].round(2)
tab_report["r [m]"] = tabular["Radius"].round(3)
tab_report["I [kg m2]"] = tabular["Moment of Inertia"].round(3)
tab_report["alpha [rad s-2]"] = tabular["Angular Acceleration"].round(3)
print(tab_report.to_latex(index=False, float_format="%.3f"))
```

```
\begin{tabular}{rrrr}
\toprule
 M [kg] &  r [m] &  I [kg m2] &  alpha [rad s-2] \\
\midrule
```

```
  0.040 &  0.030 &      0.015 &            0.791 \\
  0.060 &  0.030 &      0.016 &            1.133 \\
  0.070 &  0.015 &      0.017 &            0.618 \\
  0.070 &  0.030 &      0.016 &            1.297 \\
  0.070 &  0.045 &      0.012 &            2.673 \\
  0.080 &  0.030 &      0.016 &            1.435 \\
  0.100 &  0.030 &      0.014 &            2.092 \\
\bottomrule
\end{tabular}

/tmp/ipykernel_25126/569439553.py:6: FutureWarning: In future versions
`DataFrame.to_latex` is expected to utilise the base implementation of
`Styler.to_latex` for formatting and rendering. The arguments signature may
therefore change. It is recommended instead to use `DataFrame.style.to_latex`
which also contains additional functionality.
  print(tab_report.to_latex(index=False, float_format="%.3f"))
```