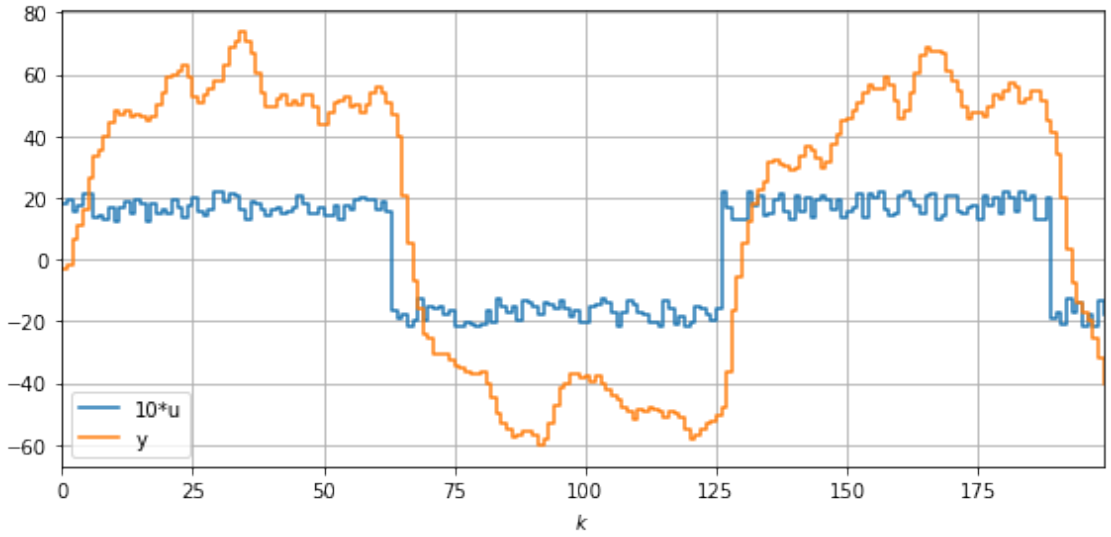
main

May 17, 2023

1 Load and View Data

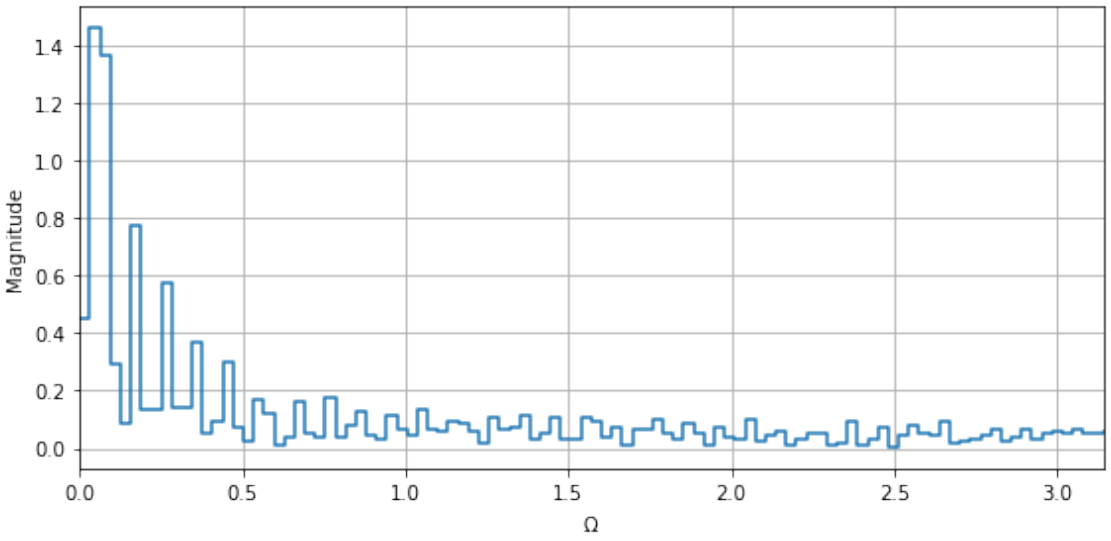
[ ]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

file = '../data.csv'
data = pd.read_csv(file, header=None, names=['u', 'y'])
N = len(data)

k = data.index.values
u = data.u.values
y = data.y.values

print('Number of data points:', N)
print(f'k in [{k[0]}, {k[-1]}]')

plt.figure(figsize=(8,4))
plt.plot(k, 10*u, label='10*u', drawstyle='steps-post')
plt.plot(k, y   , label='y'   , drawstyle='steps-post')
plt.xlim(k[0], k[-1])
plt.xlabel(r'$k$')
plt.grid()
plt.legend()
plt.tight_layout()
plt.show()
```

```
Number of data points: 200
k in [0, 199]
```



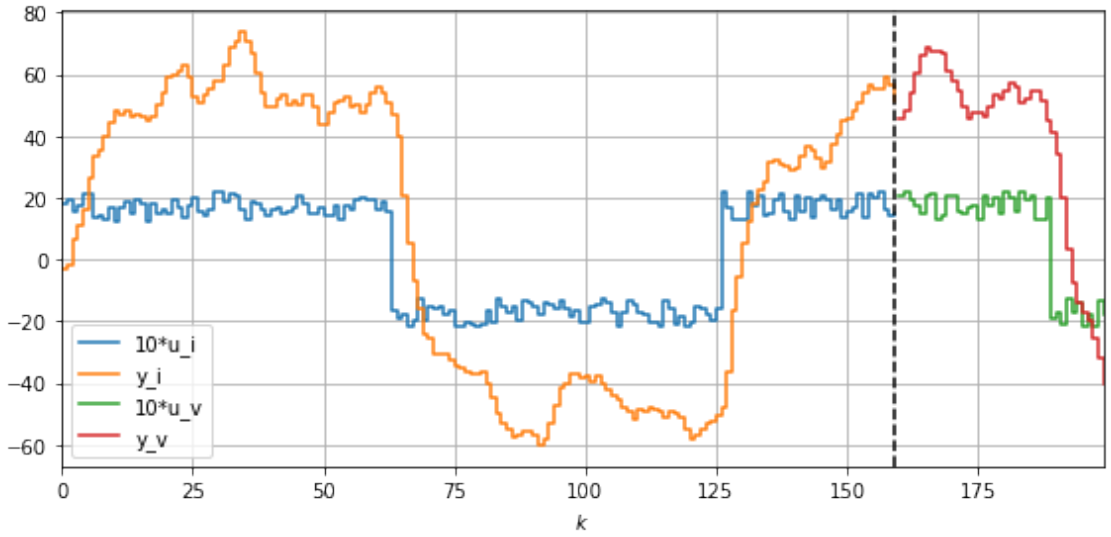
1.1 Input Fourier Transform

[ ]:
```
from scipy import fft

u_rfft = fft.rfft(u, norm='forward')
u_rfft[1:-1] = 2*u_rfft[1:-1]

u_rfft_mag = np.abs(u_rfft)
Omega = np.linspace(0, np.pi, len(u_rfft_mag))

plt.figure(figsize=(8,4))
plt.plot(Omega, u_rfft_mag, drawstyle='steps-post')
plt.xlim(Omega[0], Omega[-1])
plt.xlabel(r'$\Omega$')
plt.ylabel('Magnitude')
plt.grid()
plt.tight_layout()
plt.show()
```



1.2 Separate Identification and Validation Data

[ ]:
```
N_fold = 160

k_i = k[:N_fold]
u_i = u[:N_fold]
y_i = y[:N_fold]

k_v = k[N_fold:]
u_v = u[N_fold:]
y_v = y[N_fold:]

plt.figure(figsize=(8,4))
plt.plot(k_i, 10*u_i, label='10*u_i', drawstyle='steps-post')
plt.plot(k_i, y_i   , label='y_i'   , drawstyle='steps-post')
plt.plot(k_v, 10*u_v, label='10*u_v', drawstyle='steps-post')
plt.plot(k_v, y_v   , label='y_v'   , drawstyle='steps-post')
plt.axvline(k[N_fold-1], color='black', linestyle='--')
plt.xlim(k[0], k[-1])
plt.xlabel(r'$k$')
plt.grid()
plt.legend()
plt.tight_layout()
plt.show()
```



2 Generic Model

$$A(q)\,y[k] = \frac{B(q)}{F(q)}\,u[k-n_k+1] + \frac{C(q)}{D(q)}\,e[k]$$
$$y[k] = G(q)\,u[k-n_k+1] + H(q)\,e[k]$$
$$G(q) = \frac{B(q)}{A(q)\,F(q)} \qquad H(q) = \frac{C(q)}{A(q)\,D(q)}$$
$$A(q) = 1 - a_1\,q^{-1} - \cdots - a_{n_a}\,q^{-n_a}$$
$$B(q) = b_1\,q^{-1} + \cdots + b_{n_b}\,q^{-n_b} + +b_{n_b+1}\,q^{-n_b-1}$$
$$C(q) = 1 + c_1\,q^{-1} + \cdots + c_{n_c}\,q^{-n_c}$$
$$D(q) = 1 + d_1\,q^{-1} + \cdots + d_{n_d}\,q^{-n_d}$$
$$F(q) = 1 + f_1\,q^{-1} + \cdots + f_{n_f}\,q^{-n_f}$$

[ ]:
```
from functions import models_frame

models = models_frame()
```

3 ARX

$$y[k] = G(q)\,u[k-n_k+1] + H(q)\,e[k]$$
$$G(q) = \frac{B(q)}{A(q)} \qquad H(q) = \frac{1}{A(q)}$$
$$A(q) = 1 - a_1\,q^{-1} - \cdots - a_{n_a}\,q^{-n_a}$$
$$B(q) = b_1\,q^{-1} + \cdots + b_{n_b}\,q^{-n_b} + +b_{n_b+1}\,q^{-n_b-1}$$

[ ]:
```
from functions import arx

na_range = range(0, 3 + 1)
nb_range = range(0, 2 + 1)
nk_range = range(0, 2 + 1)

models_arx = arx(u_i, y_i, u_v, y_v, na_range, nb_range, nk_range)

models = pd.concat([models, models_arx], ignore_index=True)
```

4 ARMAX

$$y[k] = G(q)\,u[k - n_k + 1] + H(q)\,e[k]$$
$$G(q) = \frac{B(q)}{A(q)} \qquad H(q) = \frac{C(q)}{A(q)}$$
$$A(q) = 1 - a_1\,q^{-1} - \cdots - a_{n_a}\,q^{-n_a}$$
$$B(q) = b_1\,q^{-1} + \cdots + b_{n_b}\,q^{-n_b} + +b_{n_b+1}\,q^{-n_b-1}$$
$$C(q) = 1 + c_1\,q^{-1} + \cdots + c_{n_c}\,q^{-n_c}$$

[ ]:
```
from functions import armax

na_range = range(0, 3 + 1)
nb_range = range(0, 2 + 1)
nc_range = range(0, 3 + 1)
nk_range = range(0, 2 + 1)

models_armax = armax(u_i, y_i, u_v, y_v, na_range, nb_range, nc_range, nk_range)

models = pd.concat([models, models_armax], ignore_index=True)
```

5 Output Error

$$y[k] = G(q)\,u[k - n_k + 1] + H(q)\,e[k]$$
$$G(q) = \frac{B(q)}{F(q)} \qquad H(q) = 1$$
$$B(q) = b_1\,q^{-1} + \cdots + b_{n_b}\,q^{-n_b} + +b_{n_b+1}\,q^{-n_b-1}$$
$$F(q) = 1 + f_1\,q^{-1} + \cdots + f_{n_f}\,q^{-n_f}$$

[ ]:
```
from functions import oe

nb_range = range(0, 2 + 1)
nf_range = range(1, 3 + 1) # nf = 0 causa erro no pysid!
```

```
nk_range = range(0, 2 + 1)

models_oe = oe(u_i, y_i, u_v, y_v, nb_range, nf_range, nk_range)

models = pd.concat([models, models_oe], ignore_index=True)
```

6 Box-Jenkins

$$y[k] = G(q)\,u[k - n_k + 1] + H(q)\,e[k]$$
$$G(q) = \frac{B(q)}{F(q)} \qquad H(q) = \frac{C(q)}{D(q)}$$
$$B(q) = b_1\,q^{-1} + \cdots + b_{n_b}\,q^{-n_b} + +b_{n_b+1}\,q^{-n_b-1}$$
$$C(q) = 1 + c_1\,q^{-1} + \cdots + c_{n_c}\,q^{-n_c}$$
$$D(q) = 1 + d_1\,q^{-1} + \cdots + d_{n_d}\,q^{-n_d}$$
$$F(q) = 1 + f_1\,q^{-1} + \cdots + f_{n_f}\,q^{-n_f}$$

[ ]:
```
from functions import bj

nb_range = range(0, 2 + 1)
nc_range = range(0, 3 + 1)
nd_range = range(0, 3 + 1)
nf_range = range(0, 3 + 1)
nk_range = range(0, 2 + 1)

models_bj = bj(u_i, y_i, u_v, y_v, nb_range, nc_range, nd_range, nf_range,␣
 ↪nk_range)

models = pd.concat([models, models_bj], ignore_index=True)
```

7 Results

[ ]:
```
print('Number of models:', len(models.loc[models.B.notnull()]))
print('Number of fails: ', len(models.loc[models.B.isnull()]))
```

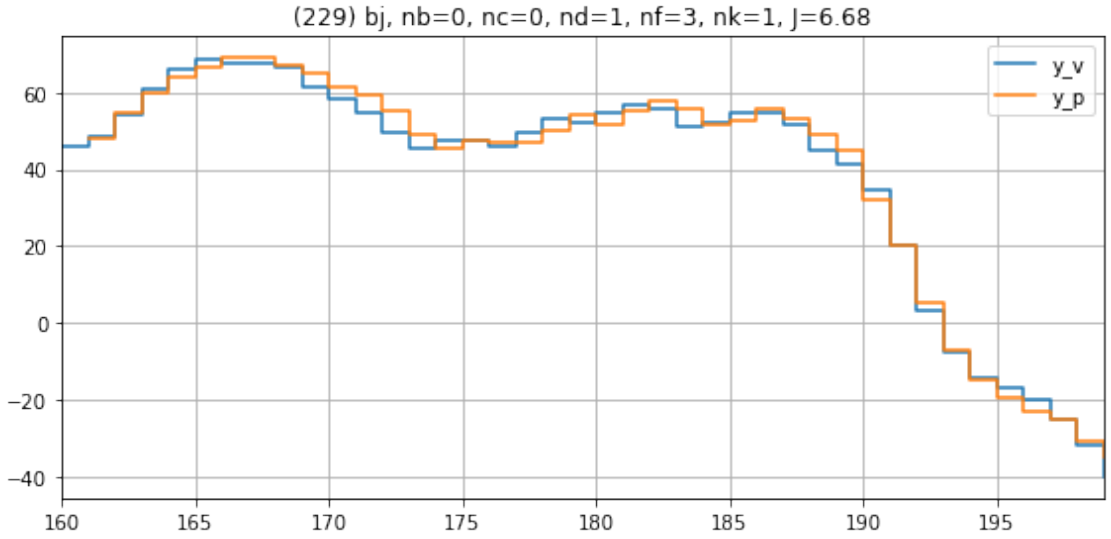
```
Number of models: 557
Number of fails:  226
```

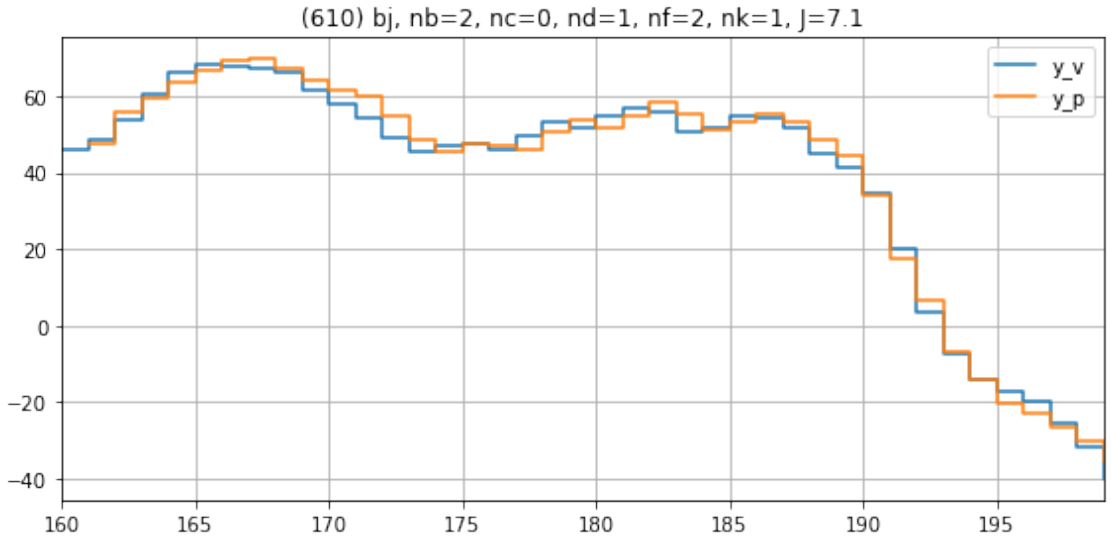
7.1 Sort by Prediction Cost

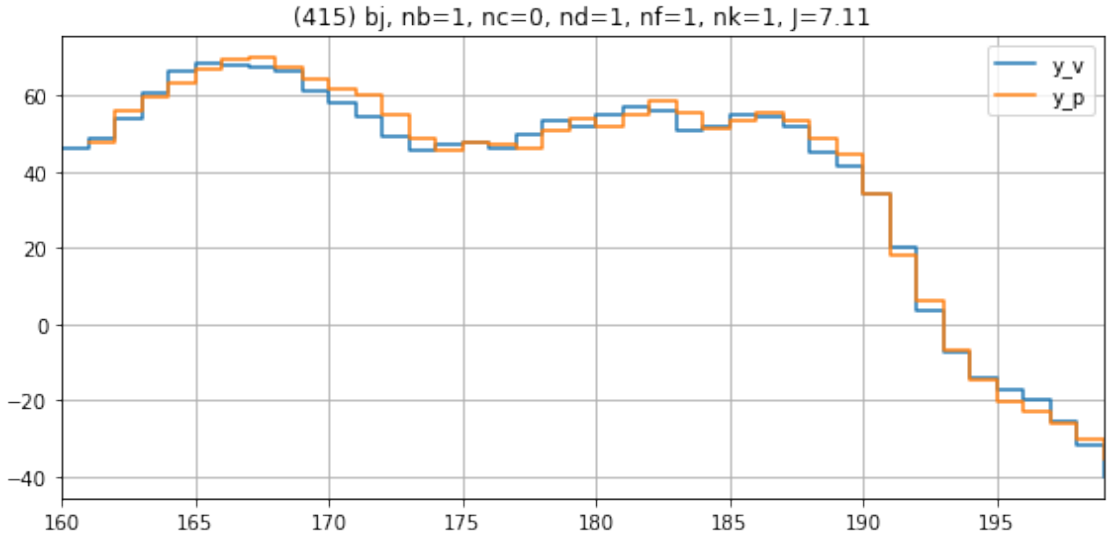
[ ]:
```
models.sort_values(by=['Jp'], inplace=True)
```

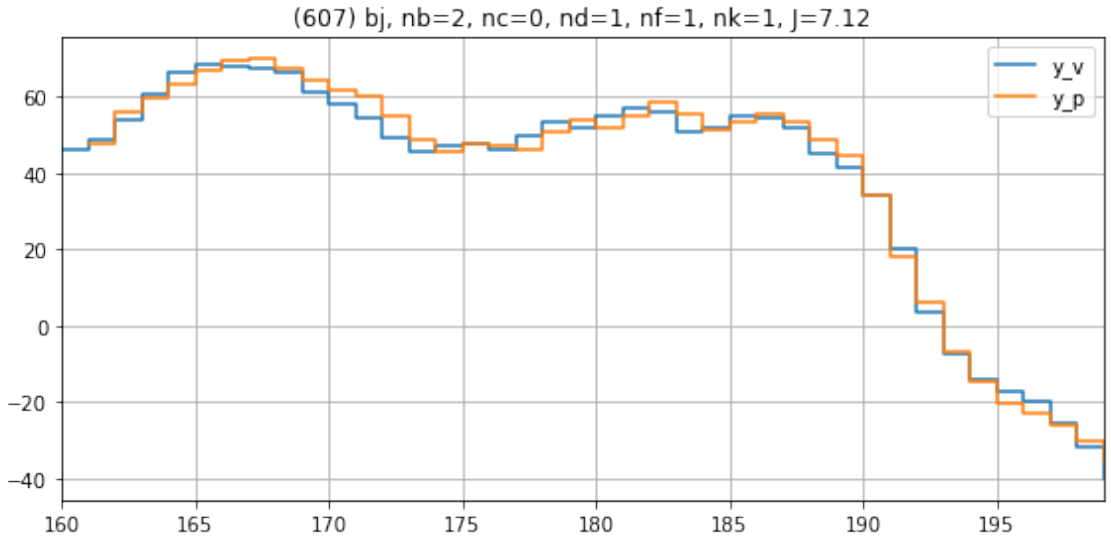
7.2 Display Predictions

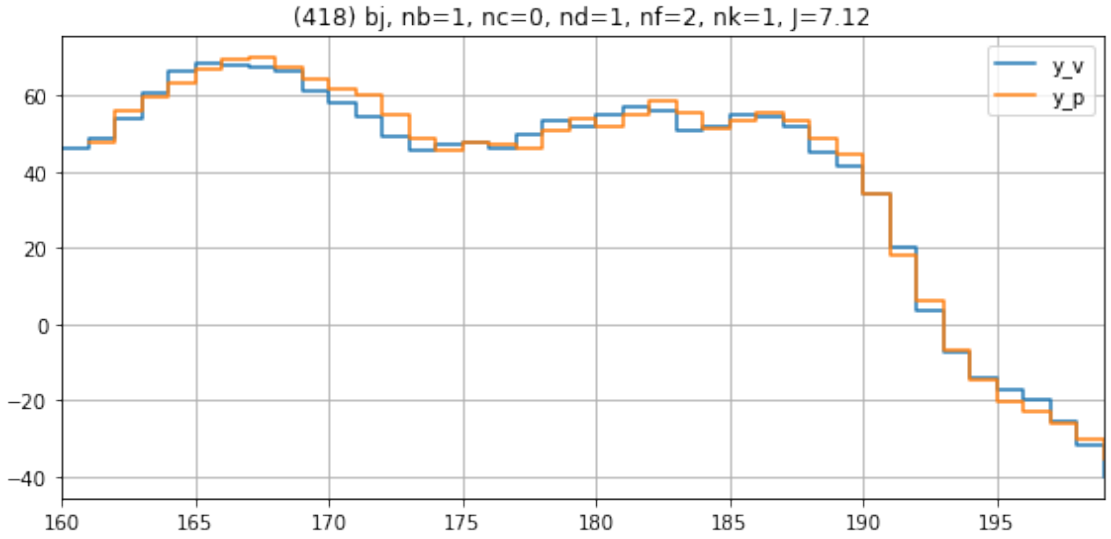
[ ]:
```
for i, (index, model) in enumerate(models.iterrows()):
  if i > 10:
    break

  if np.isnan(model.yp).any():
    continue

  if model.model == 'arx':
    title = f'({index}) {model.model}, na={model.na}, nb={model.nb}, nk={model.
↪nk}, J={model.Jp:.3g}'
  elif model.model == 'armax':
    title = f'({index}) {model.model}, na={model.na}, nb={model.nb}, nc={model.
↪nc}, nk={model.nk}, J={model.Jp:.3g}'
  elif model.model == 'oe':
    title = f'({index}) {model.model}, nb={model.nb}, nf={model.nf}, nk={model.
↪nk}, J={model.Jp:.3g}'
  elif model.model == 'bj':
    title = f'({index}) {model.model}, nb={model.nb}, nc={model.nc}, nd={model.
↪nd}, nf={model.nf}, nk={model.nk}, J={model.Jp:.3g}'
  else:
    assert(False)

  # display(model.G)

  plt.figure(figsize=(8,4))
  plt.title(title)
  plt.plot(k_v, y_v, label='y_v', drawstyle='steps-post')
  plt.plot(k_v[model.nk:], model.yp, label='y_p', drawstyle='steps-post')
  plt.xlim(k_v[0], k_v[-1])
  plt.grid()
  plt.legend()
  plt.tight_layout()
  plt.show()
```

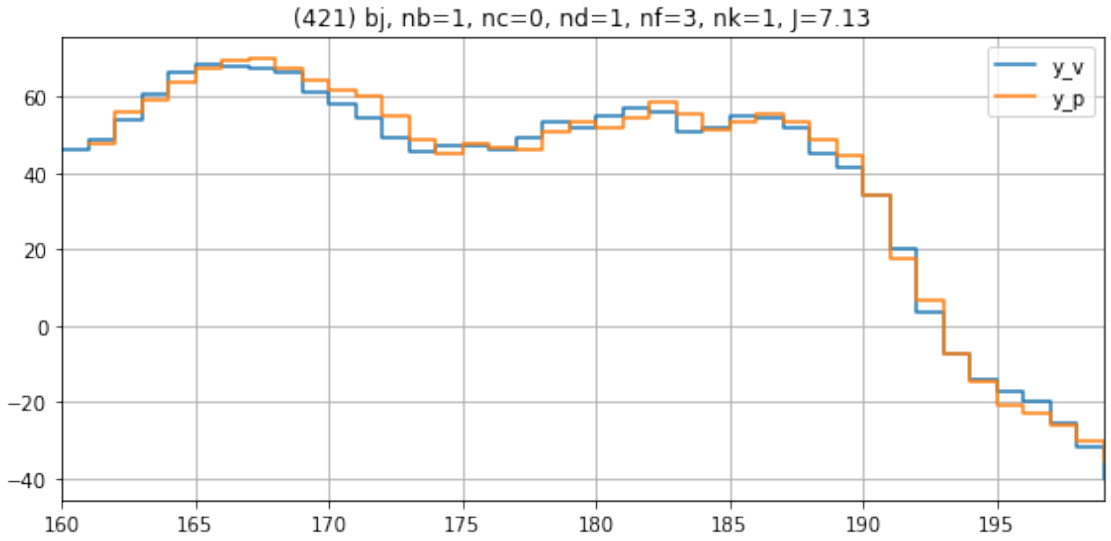

(229) bj, nb=0, nc=0, nd=1, nf=3, nk=1, J=6.68
y_v
y_p
60
40
20
0
−20
−40
160
165
170
175
180
185
190
195

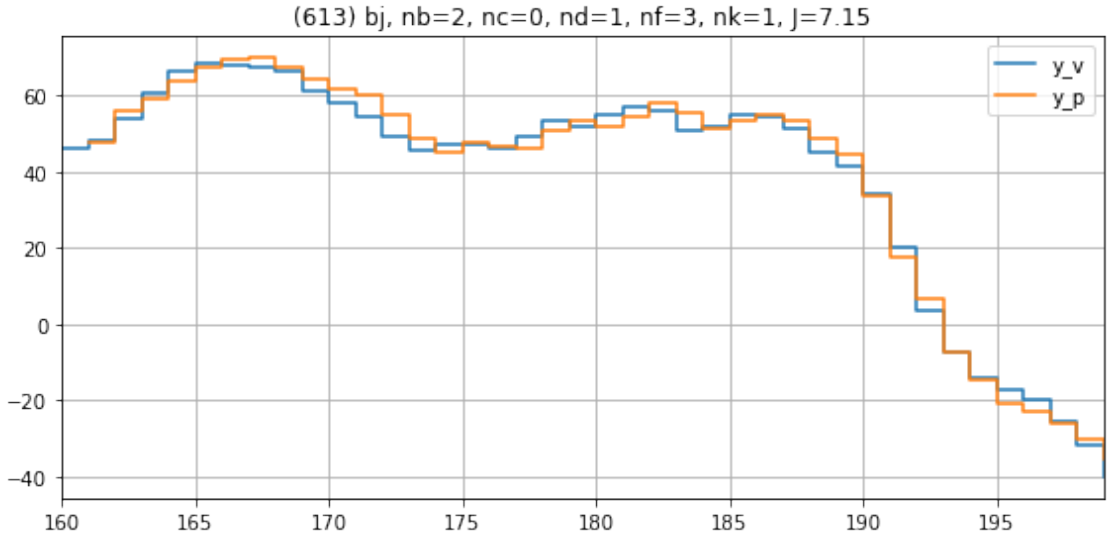

(610) bj, nb=2, nc=0, nd=1, nf=2, nk=1, J=7.1
y_v
y_p
60
40
20
0
−20
−40
160
165
170
175
180
185
190
195

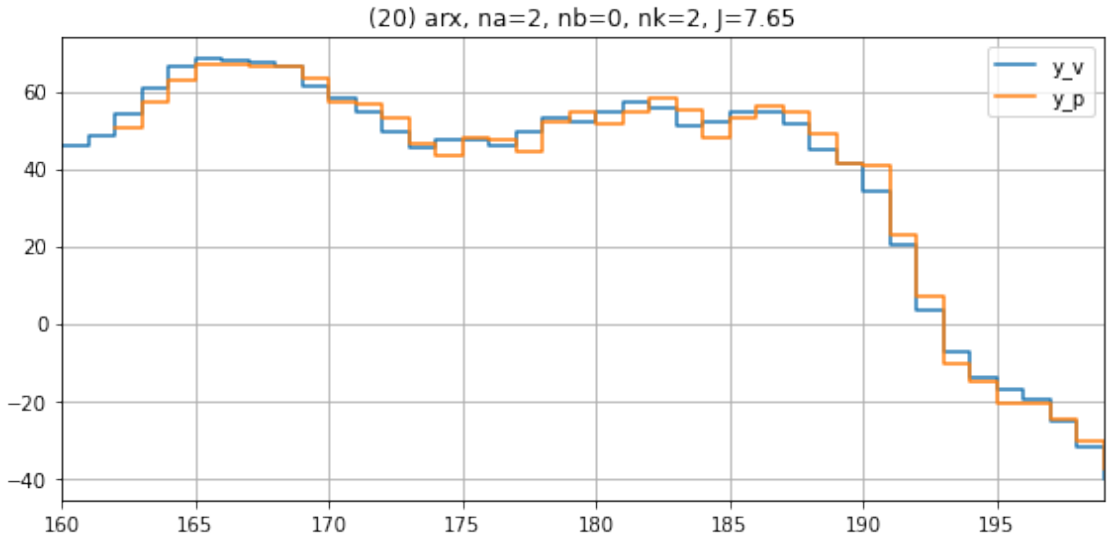

(415) bj, nb=1, nc=0, nd=1, nf=1, nk=1, J=7.11
y_v
y_p
60
40
20
0
−20
−40
160
165
170
175
180
185
190
195

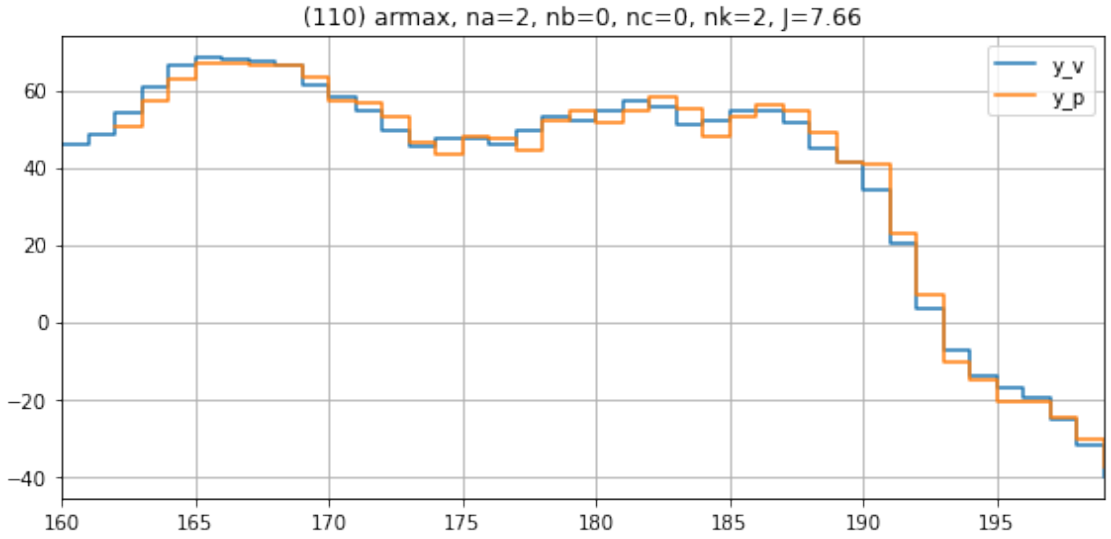

(607) bj, nb=2, nc=0, nd=1, nf=1, nk=1, J=7.12
y_v
y_p
60
40
20
0
−20
−40
160
165
170
175
180
185
190
195

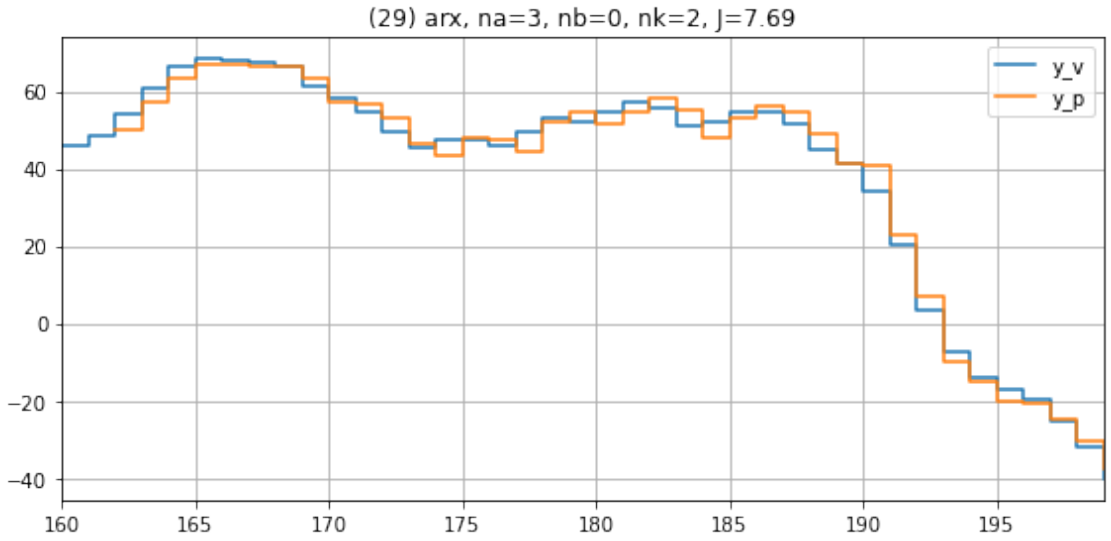

(418) bj, nb=1, nc=0, nd=1, nf=2, nk=1, J=7.12
y_v
y_p
60
40
20
0
−20
−40
160
165
170
175
180
185
190
195

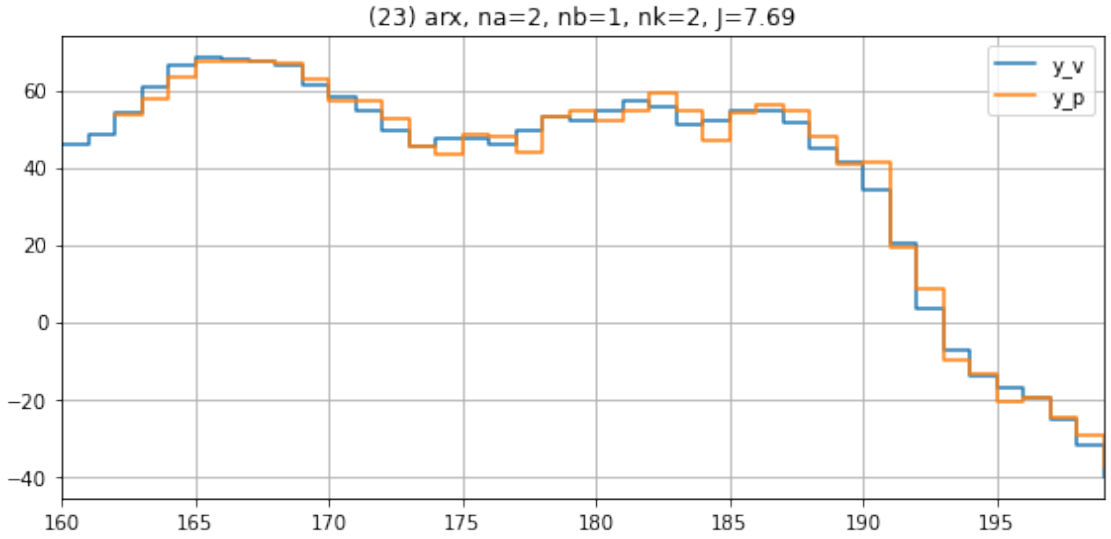

(421) bj, nb=1, nc=0, nd=1, nf=3, nk=1, J=7.13
y_v
y_p
60
40
20
0
−20
−40
160
165
170
175
180
185
190
195


(613) bj, nb=2, nc=0, nd=1, nf=3, nk=1, J=7.15
y_v
y_p
60
40
20
0
−20
−40
160
165
170
175
180
185
190
195


(20) arx, na=2, nb=0, nk=2, J=7.65
y_v
y_p
60
40
20
0
−20
−40
160
165
170
175
180
185
190
195


(110) armax, na=2, nb=0, nc=0, nk=2, J=7.66
y_v
y_p
60
40
20
0
−20
−40
160
165
170
175
180
185
190
195


(29) arx, na=3, nb=0, nk=2, J=7.69
y_v
y_p
60
40
20
0
−20
−40
160
165
170
175
180
185
190
195



7.3 Display Best ARX

[ ]:
```
display(models.loc[models.model == 'arx'][['na','nb','nk','Jp','A','B',]].
↪head(20))
```

	na	nb	nk	Jp	A	B
20	2	0	2	7.645443	[1.0, -1.3660455345977758, 0.46216420283874365]	[0.0, 0.0, 2.6625996144096495]
29	3	0	2	7.689685	[1.0, -1.3873295518220405, 0.48805419928292326…	
23	2	1	2	7.689954	[1.0, -1.4641248135410299, 0.5285400201585324]	
32	3	1	2	7.892806	[1.0, -1.4338646895641036, 0.4480849301598035,…	
35	3	2	2	8.105173	[1.0, -1.4646304544414424, 0.528125287033142, …	
26	2	2	2	8.116282	[1.0, -1.470109560817867, 0.5424885365137805]	
13	1	1	1	9.445924	[1.0, -0.8393277799197246]	
16	1	2	1	9.446921	[1.0, -0.8475073498920537]	
17	1	2	2	12.220939	[1.0, -0.8536344010412926]	
10	1	0	1	12.227166	[1.0, -0.8678590885029938]	
11	1	0	2	12.263688	[1.0, -0.8463826959667305]	
22	2	1	1	12.272655	[1.0, -1.3097104301654667, 0.41739885788678394]	
14	1	1	2	12.535729	[1.0, -0.852414481348611]	
31	3	1	1	14.254416	[1.0, -1.3524174269820388, 0.5032190693764751,…	
34	3	2	1	15.842331	[1.0, -1.3994740241334414, 0.4627146071616107,…	
25	2	2	1	16.155434	[1.0, -1.4068341173741103, 0.48261202067241576]	
19	2	0	1	16.409968	[1.0, -1.3707902290649023, 0.46379315525848536]	
28	3	0	1	18.702151	[1.0, -1.414703381673296, 0.556534746868992, -…	
9	1	0	0	53.836422	[1.0, -0.8940577163754525]	
15	1	2	0	58.178657	[1.0, -0.8385439305093618]	

```
29                      [0.0, 0.0, 2.5495944772087906]
23  [0.0, 0.0, 3.2667955231622616, -1.512659611800…
32  [0.0, 0.0, 3.2988193484523514, -1.597544061149…
35  [0.0, 0.0, 3.2526649889541677, -1.770506126503…
26  [0.0, 0.0, 3.24561533692063, -1.76727531046355…
13       [0.0, 2.5182279776811285, 2.247524574924342]
16  [0.0, 2.565934481739721, 2.624492279202261, -0…
17  [0.0, 0.0, 4.779090659243092, -0.2522758132522…
10                           [0.0, 3.9772885834983422]
11                        [0.0, 0.0, 4.478108167722055]
22       [0.0, 2.141831622134629, 0.9412691651247993]
14   [0.0, 0.0, 4.7414599587574, -0.4290815265286632]
31      [0.0, 2.1431634822971874, 0.8728783964686994]
34  [0.0, 2.155788344731435, 1.6239328839062663, -…
25  [0.0, 2.1624655631747225, 1.6108433820602714, …
19                           [0.0, 2.6558802749327906]
28                            [0.0, 2.605489877940335]
9                                 [3.2900528140628835]
15  [0.7506498902905142, 1.89369360076428, 2.17204…
```

7.4 Display Best ARMAX

[ ]:
```
display(models.loc[models.model ==␣
 ↪'armax'][['na','nb','nc','nk','Jp','A','B','C']].head(20))
```

```
    na nb nc nk         Jp                                                  A  \
110  2  0  0  2   7.655574       [1.0, -1.362122379521981, 0.4587552445700221]
146  3  0  0  2   7.746983  [1.0, -1.372107576696417, 0.48234394954692567,…
125  2  1  1  2   7.758462     [1.0, -1.4600096157027267, 0.5255264002325278]
122  2  1  0  2   7.768006     [1.0, -1.4527206948581295, 0.5192633529898458]
158  3  1  0  2   7.981943  [1.0, -1.4226945047469746, 0.43970782319280444…
170  3  2  0  2   8.308704  [1.0, -1.4503767229272535, 0.5066968102866604,…
137  2  2  1  2   8.311072      [1.0, -1.475717281818618, 0.5469116297457893]
134  2  2  0  2   8.321067     [1.0, -1.4609863882165854, 0.5343165080233506]
149  3  0  1  2   8.342984  [1.0, -0.9399801899614851, -0.208706173994137,…
161  3  1  1  2   8.469429  [1.0, -1.1844387652049873, 0.09576639519608975…
113  2  0  1  2   8.514107     [1.0, -1.2305922717569349, 0.3405767785078947]
85   1  1  0  1   9.509086                          [1.0, -0.8399456251236832]
97   1  2  0  1   9.518973                          [1.0, -0.8484592491471614]
128  2  1  2  2   9.551023    [1.0, -1.2809967354890153, 0.36654414556152054]
173  3  2  1  2   9.832684  [1.0, -0.5378590100399637, -0.8697379520070603…
131  2  1  3  2   9.932755     [1.0, -1.4900377017683388, 0.5504158109798434]
116  2  0  2  2  10.852399    [1.0, -0.9940512967773427, 0.11949413417148878]
89   1  1  1  2  11.407452                          [1.0, -0.8486142701479353]
101  1  2  1  2  11.471396                          [1.0, -0.8468051831338753]
152  3  0  2  2  11.537025  [1.0, -0.8790926145365645, -0.1998558265309171…

                                                        B  \
```

```
110                       [0.0, 0.0, 2.67814901214612]
146                      [0.0, 0.0, 2.6789063216335074]
125  [0.0, 0.0, 3.375042699009375, -1.5856254449940…
122  [0.0, 0.0, 3.373178325899714, -1.553870209586459]
158  [0.0, 0.0, 3.409553051450076, -1.6411466235616…
170  [0.0, 0.0, 3.339266589052116, -1.8938528860596…
137  [0.0, 0.0, 3.3336402291030303, -1.967736751034…
134  [0.0, 0.0, 3.323574570886201, -1.8887521811331…
149                      [0.0, 0.0, 3.249889847820225]
161  [0.0, 0.0, 3.385492016217483, -1.0482476919565…
113                      [0.0, 0.0, 3.082182901198755]
85       [0.0, 2.3600625664231103, 2.3834374176082576]
97   [0.0, 2.390729908024276, 2.711384911721081, -0…
128  [0.0, 0.0, 3.4399250148001252, -1.080417100133…
173  [0.0, 0.0, 3.384555801371594, 1.38369530339439…
131  [0.0, 0.0, 3.40971308278597, -1.7536633125344618]
116                       [0.0, 0.0, 3.52245164538883]
89   [0.0, 0.0, 3.8067868935708913, 0.5499943801358…
101  [0.0, 0.0, 3.7855378088305622, 0.5411600759363…
152                      [0.0, 0.0, 3.4578629152037212]

                                                     C
110                                                [1.0]
146                                                [1.0]
125                       [1.0, -0.011089805787230764]
122                                                [1.0]
158                                                [1.0]
170                                                [1.0]
137                        [1.0, -0.02244962850440833]
134                                                [1.0]
149                          [1.0, 0.4596703082418364]
161                         [1.0, 0.24506116219632135]
113                         [1.0, 0.19672390277132173]
85                                                 [1.0]
97                                                 [1.0]
128     [1.0, 0.1794893731060149, 0.16236729312536188]
173                           [1.0, 0.8964131465447145]
131  [1.0, -0.038179062820397985, 0.040590637656112…
116       [1.0, 0.439347639302333, 0.2817830769564706]
89                            [1.0, 0.4384234142947657]
101                           [1.0, 0.4388454181096915]
152       [1.0, 0.5681759014356293, 0.202204520470848]
```

7.5 Display Best OE

```
[ ]: display(models.loc[models.model == 'oe'][['nb','nf','nk','Jp','B','F',]].
     ↪head(20))
```

```
     nb nf nk          Jp                                                  B  \
206   2  3  2  226.888138  [0.0, 0.0, 7.978013556932356, -3.2286730261476…
197   1  3  2  226.967451  [0.0, 0.0, 7.838397199009468, -6.694009875361453]
203   2  2  2   227.08397  [0.0, 0.0, 7.711764588803071, -6.9074618014256…
194   1  2  2  227.262327  [0.0, 0.0, 7.563069203358311, -6.423786171942781]
200   2  1  2  241.402568  [0.0, 0.0, 8.08301108469978, -1.78417565469243…
202   2  2  1   249.64325  [0.0, 3.5064182856277886, -0.07824020826054166…
191   1  1  2  250.846734  [0.0, 0.0, 9.157994362787939, -4.8911579830779…
196   1  3  1  252.373055     [0.0, 3.3507417420103387, -2.7664680983782803]
205   2  3  1  256.792102  [0.0, 4.628640282915869, -8.623214903962165, 4…
188   0  3  2  261.168841                      [0.0, 0.0, 7.921348852710581]
185   0  2  2  267.121761                      [0.0, 0.0, 7.561901643498672]
199   2  1  1  271.972336  [0.0, 3.781185711341447, 2.8049375221333177, -…
204   2  3  0  274.352316  [3.1280668470184683, -5.340581221648062, 2.580…
182   0  1  2   276.99766                      [0.0, 0.0, 5.391554067951758]
187   0  3  1  277.617842                           [0.0, 2.988611484449413]
190   1  1  1  278.665287     [0.0, 4.827973256321665, -0.37190745122050206]
184   0  2  1  279.553033                           [0.0, 4.698984403191706]
181   0  1  1  280.648584                           [0.0, 4.524318877565773]
193   1  2  1  281.122009      [0.0, 4.3337745347488585, 4.2860954942594605]
195   1  3  0  285.269845          [1.5834576171801729, -1.1667113369486124]

                                                     F
206  [1.0, -0.9518156100627038, -0.3203644473814848…
197  [1.0, -1.4544706372946097, 0.4468435158751912,…
203       [1.0, -1.5244322818192362, 0.56005967515822]
194      [1.0, -1.509661698574922, 0.5468800443643043]
200                          [1.0, -0.8709600909127035]
202      [1.0, -1.6007510106966596, 0.627251005434897]
191                          [1.0, -0.8573586792852766]
196  [1.0, -2.1777873976684754, 1.625637672754031, …
205  [1.0, -2.759095117485883, 2.61038161396743, -0…
188  [1.0, -0.4573530409988488, -0.0409717896603812…
185   [1.0, -0.3205139270891644, -0.42269253608061114]
199                          [1.0, -0.8640834799035708]
204  [1.0, -2.7253048407638283, 2.578705640429255, …
182                          [1.0, -0.8153866103917702]
187  [1.0, -1.647584086767555, 1.166908619339762, -…
190                          [1.0, -0.8514087957661273]
184   [1.0, -0.8030776462543977, -0.04009386383446041]
181                          [1.0, -0.8488426199496534]
193    [1.0, 0.04900478959474314, -0.7608467552754207]
195  [1.0, -2.501677402990944, 2.1829670787314317, …
```

7.6 Display Best BJ

```
[ ]: display(models.loc[models.model ==␣
      ↪'bj'][['nb','nc','nd','nf','nk','Jp','B','C','D','F']].head(20))
```

```
    nb nc nd nf nk        Jp  \
229  0  0  1  3  1  6.675921
610  2  0  1  2  1  7.103042
415  1  0  1  1  1  7.107374
607  2  0  1  1  1  7.119668
418  1  0  1  2  1  7.122786
421  1  0  1  3  1  7.126711
613  2  0  1  3  1  7.147105
226  0  0  1  2  1  7.858919
239  0  0  2  2  2  8.720289
431  1  0  2  2  2  8.771421
428  1  0  2  1  2  8.859718
242  0  0  2  3  2  8.953137
251  0  0  3  2  2  9.044681
434  1  0  2  3  2  9.050066
620  2  0  2  1  2  9.055038
236  0  0  2  1  2  9.076879
443  1  0  3  2  2  9.136736
623  2  0  2  2  2  9.162588
440  1  0  3  1  2  9.237344
254  0  0  3  3  2  9.389977

                                                     B      C  \
229                          [0.0, 2.688141278221442]  [1.0]
610  [0.0, 2.3478282185298474, 5.096527272299687, 2…  [1.0]
415        [0.0, 2.344296218297878, 2.825520409261619]  [1.0]
607  [0.0, 2.344447984057369, 2.8318259756191324, -…  [1.0]
418         [0.0, 2.34448302850676, 2.852622937222137]  [1.0]
421      [0.0, 2.3310033884478294, 3.0402361290651934]  [1.0]
613  [0.0, 2.3308263024050233, 2.838620525444865, -…  [1.0]
226                         [0.0, 3.0115919098877106]  [1.0]
239                      [0.0, 0.0, 3.5019507656056597]  [1.0]
431  [0.0, 0.0, 3.349193072443662, -1.7775884528373…  [1.0]
428  [0.0, 0.0, 3.5482906571711608, 0.2858822969171…  [1.0]
242                      [0.0, 0.0, 3.3500091640579415]  [1.0]
251                       [0.0, 0.0, 3.565838195878382]  [1.0]
434  [0.0, 0.0, 3.331503161637364, -0.8551562311690…  [1.0]
620  [0.0, 0.0, 3.4484422221090623, 0.2726044485092…  [1.0]
236                      [0.0, 0.0, 3.567850575202136]  [1.0]
443  [0.0, 0.0, 3.376921440068028, -2.0993345353713…  [1.0]
623  [0.0, 0.0, 3.302831622129785, -1.2972424032536…  [1.0]
440  [0.0, 0.0, 3.6079100194975062, 0.2026999940729…  [1.0]
254                      [0.0, 0.0, 3.429127315563511]  [1.0]
```

```
                                                      D  \
229                          [1.0, -0.9289937416737682]
610                          [1.0, -0.9341682155923209]
415                          [1.0, -0.9343867522324996]
607                          [1.0, -0.9342305307417798]
418                           [1.0, -0.934195041218468]
421                          [1.0, -0.9369178960361715]
613                          [1.0, -0.9372026762173828]
226                          [1.0, -0.9438017519648862]
239      [1.0, -1.458155553275884, 0.5483535057734815]
431     [1.0, -1.4769712224894986, 0.5733963064298735]
428     [1.0, -1.4631480553885532, 0.5471891887933172]
242     [1.0, -1.4741934171674256, 0.5726446814107561]
251  [1.0, -1.426960883004859, 0.4619976532321713, …
434     [1.0, -1.4841070828482519, 0.5811156881875443]
620       [1.0, -1.46540220905493, 0.5601426684864359]
236     [1.0, -1.4740541947694308, 0.5494407044946709]
443  [1.0, -1.4362379085719919, 0.4606296547878398,…
623     [1.0, -1.4861249843985334, 0.5846771788274585]
440  [1.0, -1.427167998101631, 0.4510434895269983, …
254  [1.0, -1.4241186566755433, 0.44704927930510013…

                                                      F
229  [1.0, -1.5632174239302818, 0.8951586489158623,…
610     [1.0, 0.1442903549544611, -0.7839292708223434]
415                          [1.0, -0.8153741783712067]
607                          [1.0, -0.8159164961511574]
418  [1.0, -0.8078458673576828, -0.0066902219605544…
421  [1.0, -0.7291991947864063, -0.1388407532093586…
613  [1.0, -0.8169448415853702, -0.0695356924726619…
226      [1.0, -1.302072278854778, 0.4140327168562698]
239    [1.0, -0.9960096081270118, 0.11807914388348542]
431     [1.0, -1.5212473292658029, 0.5765638436863573]
428                          [1.0, -0.8662343607025677]
242  [1.0, -0.956902992202918, -0.08394568951933881…
251    [1.0, -0.9652667847846519, 0.08910273864935399]
434  [1.0, -1.1854559008583219, 0.13907771190711846…
620                          [1.0, -0.8471033232598844]
236                           [1.0, -0.875103944965068]
443     [1.0, -1.5916875036198268, 0.6365784594376858]
623      [1.0, -1.3018214862846305, 0.396225129977099]
440                           [1.0, -0.867624235105665]
254  [1.0, -0.9264964404877519, -0.1256535074300869…
```

7.7 Display Model in Class

[ ]:
```
model = models.loc[(models.model == 'arx') & (models.na == 2) & (models.nb ==
  ↪2) & (models.nk == 1)]
assert(len(model) == 1)
model = model.iloc[0]

print('G =')
display(model.G)
print('H =')
display(model.H)
print('J_p =', model.Jp)

plt.figure(figsize=(8,4))
plt.plot(k_v, y_v, label='y_v', drawstyle='steps-post')
plt.plot(k_v[model.nk:], model.yp, label='y_p', drawstyle='steps-post')
plt.xlim(k_v[0], k_v[-1])
plt.grid()
plt.legend()
plt.tight_layout()
plt.show()
```

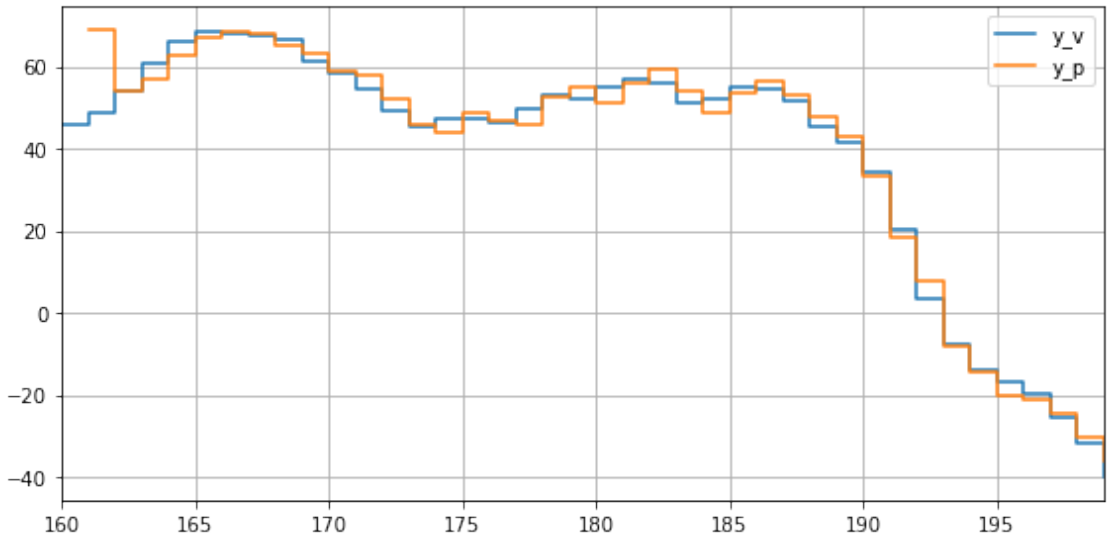
```
G =
```

$$\frac{2.162z^2 + 1.611z - 1.602}{z^3 - 1.407z^2 + 0.4826z}$$

```
H =
```

$$\frac{z^2}{z^2 - 1.407z + 0.4826}$$

```
J_p = 16.155434149353134
```


y_v
y_p
60
40
20
0
−20
−40
160
165
170
175
180
185
190
195