In [1]:
```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

Ένα απλό State Space model - Εκτίμηση θέσης κινούμενου οχήματος

- Θεωρούμε μονοδιάστατο κόσμο (1D κίνηση).
- Υποθέτουμε ότι το όχημα απομακρύνεται με σταθερή (ή σχεδόν σταθερή) επιτάχυνση.
- Θεωρούμε ότι υπάρχουν ενδείξεις από ένα GPS με κάποιο περιθώριο σφάλματος.

In [62]:
```
N = 201
dt = 4 # s
v0 = 10 # m/s
x0 = 0 # m
a0 = 0.5 # m/s^2
sigma_gps = 20.0 # m καθορίζει το περιθώριο αστοχίας του GPS
sigma_a = 0.1 # m/s^2 καθορίζει την απόκλιση του μοντέλου από την υπόθεση σταθε
ρής επιτάχυνσης
```

Δυναμικό μοντέλο του συστήματος

$$x_n = x_{n-1} + v_{n-1}\Delta t + \frac{1}{2}a_{n-1}\Delta t^2$$
$$v_n = v_{n-1} + a_{n-1}\Delta t$$
$$a_n = a_{n-1}$$

In [63]:
```
np.random.seed(1)
a = a0 + sigma_a * np.random.randn(N)
v = [v0]
x = [x0]
for i in range(1, N):
    v.append(v[i-1] + a[i-1]*dt)
    x.append(x[i-1] + v[i-1]*dt + 0.5*a[i-1]*dt**2)
z = x + sigma_gps * np.random.randn(N)
z[0] = np.nan
```

In [64]:
```
dataset = pd.DataFrame()
dataset['v'] = v
dataset['x'] = x
dataset['a'] = a
dataset['z'] = z
```

In [65]:
```
dataset
```

Out[65]:

	v	x	a	z
0	10.000000	0.000000	0.662435	NaN
1	12.649738	45.299476	0.438824	34.053368
2	14.405036	99.409024	0.447183	138.506585
3	16.193767	160.606629	0.392703	133.967595
4	17.764579	228.523321	0.586541	193.309550
...	...	...	...	...
196	409.383907	162964.744904	0.560232	162983.157206
197	411.624835	164606.762388	0.542028	164606.055029
198	413.792948	166257.597953	0.581095	166299.810054
199	416.117328	167917.418505	0.604444	167891.287823
200	418.535105	169586.723372	0.459912	169588.250981

201 rows × 4 columns

In [71]:
```
# Παράμετροι του state-space model. Καθορίζονται από την ακρίβεια του GPS
# καθώς και την καταλληλότητα του μοντέλου να περιγράψει το πραγματικό σύστημα

alpha = 0.4
beta = 0.1
gamma = 0.001

# Εκτιμήσεις για την αρχική κατάσταση του συστήματος (για τον χρόνο t_0)
x00 = x[0]
v00 = v[0]
a00 = a[0]
```

Πρόβλεψη της κάτασταση της του συστήματος για την επόμενη χρονική στιγμή

$$\hat{x}_{n,n-1} = \hat{x}_{n-1,n-1} + \hat{v}_{n-1,n-1}\Delta t + \frac{1}{2}\hat{a}_{n-1,n-1}\Delta t^2$$
$$\hat{v}_{n,n-1} = \hat{v}_{n-1,n-1} + \hat{a}_{n-1,n-1}\Delta t$$
$$\hat{a}_{n,n-1} = \hat{a}_{n-1,n-1}$$

Αναπροσαρμογή της πρόβλεψης με βάση τη μέτρηση του GPS

$$\hat{x}_{n,n} = \hat{x}_{n,n-1} + \alpha(z_n - \hat{x}_{n,n-1})$$
$$\hat{v}_{n,n} = \hat{v}_{n,n-1} + \beta\frac{z_n - \hat{x}_{n,n-1}}{\Delta t}$$
$$\hat{a}_{n,n} = \hat{a}_{n,n-1} + \gamma\frac{2(z_n - \hat{x}_{n,n-1})}{\Delta t^2}$$

In [76]:
```
hat_x_n_nm1 = [np.nan]
hat_v_n_nm1 = [np.nan]
hat_a_n_nm1 = [np.nan]
hat_x_n_n = [x00]
hat_v_n_n = [v00]
hat_a_n_n = [a00]

for n in range(1, N):
    # Προβλέψη για θέση, ταχυτητά και επιτάχυνση για την επόμενη χρονική στιγμή
    hat_x_n_nm1.append(hat_x_n_n[n-1] + hat_v_n_n[n-1] * dt + 0.5 * hat_a_n_n[n
-1]*dt**2)
    hat_v_n_nm1.append(hat_v_n_n[n-1] + hat_a_n_n[n-1] * dt)
    hat_a_n_nm1.append(hat_a_n_n[n-1])
    # Αναπροσαρμογή των εκτιμήσεων λαμβάνοντας την ένδειξη του GPS
    hat_x_n_n.append(hat_x_n_nm1[n] + alpha * (z[n]-hat_x_n_nm1[n]))
    hat_v_n_n.append(hat_v_n_nm1[n] + beta * (z[n]-hat_x_n_nm1[n]) / dt)
    hat_a_n_n.append(hat_a_n_nm1[n] + gamma * 2 * (z[n]-hat_x_n_nm1[n]) / dt**
2)

dataset['hat_a_n_(n-1)'] = hat_a_n_nm1
dataset['hat_v_n_(n-1)'] = hat_v_n_nm1
dataset['hat_x_n_(n-1)'] = hat_x_n_nm1
dataset['hat_a_n_n'] = hat_a_n_n
dataset['hat_v_n_n'] = hat_v_n_n
dataset['hat_x_n_n'] = hat_x_n_n
```

In [73]:
```
dataset
```

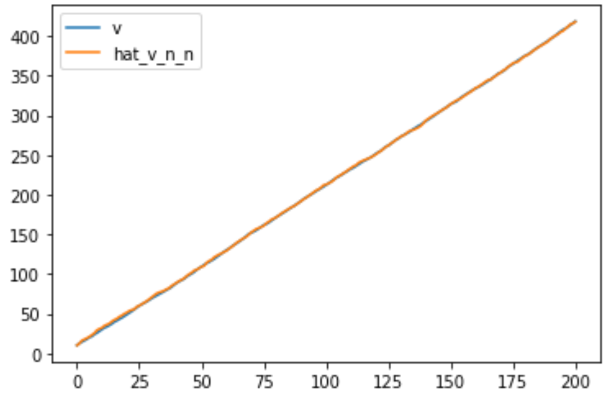
Out[73]:

	v	x	a	z	hat_a_n_(n-1)	hat_v_n_(n-1)	hat_x_n_(n-1)	hat_a_
0	10.000000	0.000000	0.662435	NaN	NaN	NaN	NaN	0.662
1	12.649738	45.299476	0.438824	34.053368	0.662435	12.649738	45.299476	0.661
2	14.405036	99.409024	0.447183	138.506585	0.661029	15.012701	95.563605	0.666
3	16.193767	160.606629	0.392703	133.967595	0.666397	18.751862	182.417070	0.660
4	17.764579	228.523321	0.586541	193.309550	0.660340	20.181987	238.482503	0.654
...	...	...	...	...	...	...	...	
196	409.383907	162964.744904	0.560232	162983.157206	0.515449	408.352976	162941.651603	0.520
197	411.624835	164606.762388	0.542028	164606.055029	0.520637	411.473164	164599.981404	0.521
198	413.792948	166257.597953	0.581095	166299.810054	0.521396	413.710590	166253.082045	0.527
199	416.117328	167917.418505	0.604444	167891.287823	0.527237	416.987740	167935.506310	0.521
200	418.535105	169586.723372	0.459912	169588.250981	0.521710	417.969118	169585.521707	0.522

201 rows × 10 columns

In [74]:
```
dataset[['v', 'hat_v_n_n']].plot()
```

Out[74]: <matplotlib.axes._subplots.AxesSubplot at 0x7f3c09804590>



In [75]:
```
plt.figure(figsize=(15,8))
plt.plot(dataset['x'] - dataset['z'], label=r'$x_n - z_n$', c='C0')
plt.plot(dataset['x'] - dataset['hat_x_n_n'], label=r'$x_n - \hat x_{n,n}$', c=
'C3')
plt.plot(dataset['x'] - dataset['hat_x_n_(n-1)'], label=r'$x_n - \hat x_{n,n-
1}$', c='C4', ls='--',alpha=0.3)
plt.legend()
```

Out[75]: <matplotlib.legend.Legend at 0x7f3c098e5750>