



# گزارش پایانی از سخت افزار

دانشکده مهندسی کامپیوتر، دانشگاه صنعتی شریف

آرین احدی نیا  
مصطفی اوجاقي  
امیر سپهر پورفناد

استاد درس: جناب آقای دکتر اجلالی  
دستیاران آموزشی: جناب آقای فصحتی، سرکار خانم رضازاد

پاییز ۱۴۰۱

## فهرست مطالب

۳	فیلتر جهت مجویک
۵	تخمین مکان
۱۰	جمع بندی

## فیلتر جهت مجویک

الگوریتم مجویک که توسط سباستین مجویک ارائه شده است، برای ژيروسکوپ و شتابسنج‌های سه محوره قابل اعمال است.

این فیلتر از یک نمایش چهارگانه جهت برای توصیف ماهیت جهت‌گیری‌ها در فضای سه بعدی استفاده می‌کند و مشمول تکنیک‌های مرتبط با نمایش زاویه اولیه و اویلر نمی‌شود و اجازه می‌دهد تا داده‌های شتابسنج و مغناطیس‌سنج در یک الگوریتم گرادیان-نزولی به صورت تحلیلی مشتق‌شده و بهینه‌شده برای محاسبه جهت خطای اندازه‌گیری ژيروسکوپ استفاده شوند.

جنبه‌های خلاقانه این فیلتر عبارتند از

۱. یک پارامتر منفرد قابل تنظیم که توسط ویژگی‌های سیستم‌های قابل مشاهده تعریف می‌شود.
۲. یک الگوریتم گرادیان نزولی که به صورت تحلیلی مشتق شده و بهینه شده است که عملکرد را در نرخ نمونه‌برداری پایین ممکن می‌سازد.
۳. الگوریتم جبران اعوجاج مغناطیسی آنلاین
۴. جبران رانش سوگیری ژيروسکوپ

توجه کنید که سوگیری زمین نسبت به سنسور را اگر یک بردار چهار بعدی  $\mathbf{q}_{\omega,t} = [q_w \ q_x \ q_y \ q_z]$  در زمان  $t$  در نظر بگیریم، می‌توانیم با استفاده از دادگان زاویه‌ای  $\omega = [\omega_x \ \omega_y \ \omega_z]$  بدست آمده در بازه زمانی  $\Delta t$  و انتگرال‌گیری نسبت به مشتق  $\dot{\mathbf{q}}_t = \frac{1}{\Delta t}(\mathbf{q}_t - \mathbf{q}_{t-1})$  عبارت

$$\begin{aligned}\mathbf{q}_{\omega,t} &= \mathbf{q}_{t-1} + \dot{\mathbf{q}}_{\omega,t} \Delta t \\ &= \mathbf{q}_{t-1} + \frac{1}{\Delta t}(\mathbf{q}_{t-1}^T \omega_t) \Delta t\end{aligned}$$

مقادیر آنها را بدست آوریم.

اگر فرض کنیم که زمین یک مرجع ثابت باشد که  $\mathbf{d}^E = [d_x \ d_y \ d_z]$  و سرعت نسبت به آن به صورت

$$S_s = \begin{bmatrix} 0 & s_x & s_y & s_z \end{bmatrix}$$

اندازه‌گیری شود، می‌توانیم تابع هدف را به صورت

$$f(\mathbf{q}, {}^E\mathbf{d}, S_s) = \mathbf{q}^* {}^E\mathbf{d} \mathbf{q} - S_s$$

$$= \begin{bmatrix} 2d_x(\frac{1}{4} - q_y^2 q_z^2) + 2d_y(q_w q_z + q_x q_y) + 2d_z(q_x q_z - q_w q_y) - s_x \\ 2d_x(q_x q_y - q_w q_z) + 2d_y(\frac{1}{4} - q_x^2 q_z^2) + 2d_z(q_w q_x + q_y q_z) - s_y \\ 2d_x(q_w q_y + q_x q_z) + 2d_y(q_y q_z - q_w q_x) + 2d_z(\frac{1}{4} - q_x^2 q_y^2) - s_z \end{bmatrix}$$

بنویسیم به نحوی که جواب مساله با کمینه‌سازی این تابع حاصل آید. برای کمینه‌سازی این جواب از الگوریتم گرادیان- نزولی استفاده می‌کنیم.

در فضای سه بعدی، طبق قضیه چرخش اوایلر، هر چرخش یا دنباله‌ای از چرخش یک جسم صلب یا سیستم مختصات حول یک نقطه ثابت، معادل یک چرخش منفرد در یک زاویه معین [۴] در مورد یک نقطه ثابت است. محور (به نام محور اوایلر) که از نقطه ثابت می‌گذرد. محور اوایلر معمولاً با یک بردار واحد نشان داده می‌شود. بنابراین، هر چرخش در سه بعدی را می‌توان به صورت ترکیبی از یک بردار  $u$  و یک اسکالر [۴] نشان داد. چهارگانه‌ها راه ساده‌ای را برای رمزگذاری این نمایش محور-زاویه در چهار عدد ارائه می‌دهند و می‌توان از آنها برای اعمال (محاسبه) چرخش متناظر به بردار موقعیت  $(x, y, z)$  استفاده کرد که نشان دهنده یک نقطه نسبت به مبدا در فضای سه بعدی است.

بنابراین گام‌های این الگوریتم عبارت است از

۱. در ابتدا می‌بایست شتاب‌سنج را در یک موقعیت مکانی و جهتی مشخص که آن را می‌دانیم قرار داد.
۲. داده‌ها را از شتاب‌سنج می‌خوانیم.
۳. در این گام با استفاده از یکی از الگوریتم‌هایی که در ادامه مطرح می‌شود جهت (orientation) را بدست می‌آوریم. جهت را به فرم quaternion نمایش می‌دهیم. پس از پایان الگوریتم درباره‌ی quaternion توضیحاتی داده خواهد شد.
۴. با استفاده از جهت فعلی بدست آمده یک ماتریس چرخش می‌سازیم (rotation). (matrix) این ماتریس مبدا داده‌های شتاب‌سنج را که نسبت به خود این دستگاه هستند را به مبدا دنیای حقیقی (۳ محور مکان) تبدیل می‌کند. در فرمول زیر  $R$  همان ماتریس چرخش است که در شتابی که از دستگاه خوانده می‌شود ضرب می‌شود. سمت چپ تساوی نیز شتاب‌های خوانده شده از دستگاه در دستگاه مختصات مبدا است.
۵. با استفاده از انتگرال گیری روی داده‌های تبدیل شده سرعت دستگاه را بدست می‌آوریم.

۶. دوباره با استفاده از انتگرال گیری روی سرعت بدست آمده شتاب دستگاه را بدست می آوریم.

۷. برو به گام ۲

## تخمین مکان

الگوریتم فوق ساده ترین راه استفاده از خروجی شتاب سنج برای بدست آوردن مکان است. با این حال، همه سنسورها دارای سوگیری هستند، بنابراین وقتی خروجی محاسبه می شود، در تخمین های سرعت، موقعیت و جهت گیری ممکن است دچار خطا شویم (زیرا آنها تخمین هستند و اندازه گیری نیستند). به طور خاص در گام تخمین برای محاسبه جهت گیری روش های گوناگونی وجود دارد اما پایه تمامی آنها بدست آوردن سرعت های زاویه ای است. الگوریتم های مختلف سعی دارند تا با استفاده از پارامترهای دیگر که شتاب سنج خروجی می دهد به اصلاح خطاهای موجود در محاسبه این سرعت های زاویه ای بپردازند. برای مثال، فیلتر Madgwick در تخمین جهت گیری بهتر از انتگرال گیری از خروجی شتاب سنج عمل می کند، زیرا فیلتر Madgwick از خروجی بردار جاذبه های شتاب سنج برای حذف خطاهای محاسبه جهت استفاده می کند. در این گزارش ما از الگوریتم Madgwick استفاده می کنیم و در گام های بعد پروژه به بررسی الگوریتم های Mahony و Extended Kalman Filter و انتگرال گیری می پردازیم.

نسخه اولیه کد عبارت است از

```
1 import time
2 from math import radians
3
4 import numpy as np
5 import quaternion
6 from ahrs.filters import Madgwick
7 from magno_gy import gy801
8
9
10 sensors = gy801()
11 gyro = sensors.gyro
12 accel = sensors.accel
13 compass = sensors.compass
14 barometer = sensors.baro
15
16 def read_accel():
17     return accel.getX(), accel.getY(), accel.getZ()
18
19 def read_gyro():
20     return radians(gyro.getX()), radians(gyro.getY()), radians(gyro.getZ())
21
22 def read_magnet():
23     return compass.getX(), compass.getY(), compass.getZ()
24
25 madgwick_wo_magnet = Madgwick()
```

```

26 madgwick_w_magnet = Madgwick()
27 Q_w_magnet = [1., 0., 0., 0.]
28
29 t = 0
30 v_wo = np.array([0., 0., 0.])
31 v_w = np.array([0., 0., 0.])
32
33 x_wo = np.array([0., 0., 0.])
34 x_w = np.array([0., 0., 0.])
35
36
37 def calcualte_rotation_matrix(Q):
38     np_quaternion = np.quaternion(Q[0], Q[1], Q[2], Q[3])
39     return quaternion.as_rotation_matrix(np_quaternion)
40
41 class IMUData:
42     def __init__(self, gyr, acc, mag) -> None:
43         self.gyr = gyr
44         self.acc = acc
45         self.mag = mag
46
47 def get_imu_data() -> IMUData:
48     acc = read_accel()
49     gyr = read_gyro()
50     mag = read_magnet()
51
52     return IMUData(gyr, acc, mag)
53
54 class AccelartionCalculator:
55     def __init__(self):
56         self.q_prev = [1., 0., 0., 0.]
57
58     def get_q(self, imu_data: IMUData, dt):
59         pass
60
61     def get_acceleration(self, imu_data: IMUData, dt, a0=np.array([0, 0,
62 0])):
63         q = self.get_q(imu_data, dt)
64         rotation_matrix = calcualte_rotation_matrix(q)
65         a = np.dot(rotation_matrix, imu_data.acc) - a0
66
67         self.q_prev = q
68         return a
69
70 class AccelartionCalculatorWithOutMagnet(AccelartionCalculator):
71     def get_q(self, imu_data: IMUData, dt):
72         q = madgwick_w_magnet.updateIMU(self.q_prev, gyr=imu_data.gyr, acc=
imu_data.acc, dt=dt)
73         return q
74
75 class AccelartionCalculatorWithMagnet(AccelartionCalculator):
76     def get_q(self, imu_data: IMUData, dt):
77         q = madgwick_w_magnet.updateMARG(self.q_prev, gyr=imu_data.gyr, acc
=imu_data.acc, mag=imu_data.mag, dt=dt)
78         return q

```

```

79
80 a_calculator_w = AccelartionCalculatorWithMagnet()
81 a_calculator_wo = AccelartionCalculatorWithOutMagnet()
82
83 prev_time = time.time()
84 imu_data = get_imu_data()
85 dt = time.time() - prev_time
86 prev_time = time.time()
87 a0_w = a_calculator_w.get_acceleration(imu_data, dt)
88 a0_wo = a_calculator_wo.get_acceleration(imu_data, dt)
89
90
91 prev_time = time.time()
92 while True:
93     t += 1
94     curr_time = time.time()
95     dt = curr_time - prev_time
96     prev_time = curr_time
97     imu_data = get_imu_data()
98
99     a_wo = a_calculator_wo.get_acceleration(imu_data, dt, a0=a0_wo)
100    a_w = a_calculator_w.get_acceleration(imu_data, dt, a0=a0_w)
101
102    v_wo += a_wo * dt
103    v_w += a_w * dt
104
105    x_wo += v_wo * dt
106    x_w += v_w * dt
107
108    if t % 100 == 0:
109        print(f'acceleration without magnometer:\t\t{a_wo}')
110        print(f'acceleration with magnometer:\t\t{a_w}')
111        print('*'*30)
112        print(f'veLOCITY without magnometer:\t\t{v_wo}')
113        print(f'veLOCITY with magnometer:\t\t{v_w}')
114        print('*'*30)
115        print(f'location without magnometer:\t\t{x_wo}')
116        print(f'location with magnometer:\t\t{x_w}')
117
118    print('\n'*4)

```

و خروجی آن برابر خواهد بود با

```

1 acceleration without magnometer:    [-4.25149814e-03 -4.95737249e-02
   9.42624897e+00]
2 acceleration with magnometer:      [-0.04666684 -0.82391009  9.39018848]
3 *****
4 velocity without magnometer:        [-0.08738802 -0.02829634 11.45156396]
5 velocity with magnometer:           [-0.1709201  -0.47476062 11.4373586 ]
6 *****
7 location without magnometer:         [-0.0767284  -0.01584075  7.01596472]
8 location with magnometer:            [-0.12772053 -0.19176933  7.01128514]
9
10 acceleration without magnometer:    [-0.02518617 -0.02588419  9.42638894]
11 acceleration with magnometer:       [ 0.1183487  -1.54551333  9.2981447 ]
12 *****

```

```

13 velocity without magnometer: [-0.11273455  0.06517335  23.07376832]
14 velocity with magnometer:   [-0.13394683 -1.81403115  22.97855815]
15 *****
16 location without magnometer: [-1.99901281e-01  1.35208854e-02
    2.83287529e+01]
17 location with magnometer:   [-0.33542242 -1.50777399  28.268977  ]
18
19 acceleration without magnometer: [-2.83333815e-02 -2.27021052e-03
    9.46571874e+00]
20 acceleration with magnometer:   [ 0.1991406  -2.03568396  9.24213033]
21 *****
22 velocity without magnometer: [-0.13622949 -0.20122665  34.72070666]
23 velocity with magnometer:   [ 0.05955209 -4.32003195  34.351887  ]
24 *****
25 location without magnometer: [-3.55068346e-01 -4.62302547e-02
    6.40790289e+01]
26 location with magnometer:   [-0.38755003 -5.21982079  63.75090291]
27
28 acceleration without magnometer: [ 2.14537055e-03 -8.36774444e-03
    9.42645416e+00]
29 acceleration with magnometer:   [ 0.2035248  -2.13368564  9.17954661]
30 *****
31 velocity without magnometer: [-0.15878657 -0.34168176  46.41142582]
32 velocity with magnometer:   [ 0.30219398 -7.08084202  45.70950965]
33 *****
34 location without magnometer: [ -0.53768706  -0.35373549
    114.40316292]
35 location with magnometer:   [ -0.15114503 -12.25904527
    113.41877558]
36
37 acceleration without magnometer: [ 1.29001854e-03 -2.65921315e-03
    9.38816218e+00]
38 acceleration with magnometer:   [ 0.1189121  -2.0876348  9.15233515]
39 *****
40 velocity without magnometer: [-0.29744507 -0.48793365  58.08000237]
41 velocity with magnometer:   [ 0.33816309 -9.80783865  57.05681006]
42 *****
43 location without magnometer: [ -0.84045521  -0.8003985
    179.03378125]
44 location with magnometer:   [ 0.25672646 -22.64583887  176.996302
    ]
45
46 acceleration without magnometer: [-0.02331939 -0.03125536  9.46459583]
47 acceleration with magnometer:   [ 0.01744526 -2.0486378  9.24028538]
48 *****
49 velocity without magnometer: [-0.32350558 -0.51723034  69.73941834]
50 velocity with magnometer:   [ 0.42745446 -12.3615718
    68.43273926]
51 *****
52 location without magnometer: [ -1.22231378  -1.41875304
    258.01989447]
53 location with magnometer:   [ 0.7415673  -36.35719339
    254.54020762]
54
55 acceleration without magnometer: [-3.23846762e-03 -5.46642539e-03
    9.38808234e+00]

```



```

56 acceleration with magnometer:      [-0.04872244 -1.92553817  9.18836544]
57 *****
58 velocity without magnometer:      [-0.34674657 -0.54121791 81.44250122]
59 velocity with magnometer:         [ 0.40057772 -14.79097339 79.8814177
60 ]
61 *****
62 location without magnometer:       [ -1.64033198 -2.01828867
351.81844505]
63 location with magnometer:          [ 1.28347787 -53.16669974
346.56780088]

```

همانطور که مشاهده میکنید بردار عمودی آن دارای شتاب نزدیک به ۱۰ هست که همان شتاب گرانش کره زمین است. در گام بعد این شتاب را از محاسبات حذف میکنیم. برای این کار کافی است تا خطوط ۹۹ و ۱۰۰ از کد را تغییر دهیم

```

1 a_wo = a_calculator_wo.get_acceleration(imu_data, dt)
2 a_w = a_calculator_w.get_acceleration(imu_data, dt)

```

خروجی برابر خواهد شد با

```

1 acceleration without magnometer:  [ 0.21294515 -0.05107898  0.46243406]
2 acceleration with magnometer:     [ 0.17275839 -0.81592309  0.42991005]
3 *****
4 velocity without magnometer:      [ 0.18002359 -0.07464414  0.52304873]
5 velocity with magnometer:         [ 0.09035456 -0.51137019  0.50877135]
6 *****
7 location without magnometer:       [ 0.08789094 -0.04398274  0.32123195]
8 location with magnometer:          [ 0.03570646 -0.21272872  0.31651788]
9
10 acceleration without magnometer:  [ 0.22391228 -0.09468307  0.50056942]
11 acceleration with magnometer:     [ 0.33809077 -1.66570507  0.35887181]
12 *****
13 velocity without magnometer:      [ 0.42873573 -0.03035206  1.05489486]
14 velocity with magnometer:         [ 0.38949088 -1.89878573  0.95883246]
15 *****
16 location without magnometer:       [ 0.46382656 -0.10506909  1.29079559]
17 location with magnometer:          [ 0.31357678 -1.59880393  1.23082621]
18
19 acceleration without magnometer:  [ 0.15097011 -0.0447729  0.42603422]
20 acceleration with magnometer:     [ 0.36940456 -2.12815785  0.18960435]
21 *****
22 velocity without magnometer:      [ 0.68030172 -0.09790213  1.58801106]
23 velocity with magnometer:         [ 0.86183677 -4.22533187  1.26035704]
24 *****
25 location without magnometer:       [ 1.14623013 -0.18067855  2.91013427]
26 location with magnometer:          [ 1.07312008 -5.29330579  2.60402268]
27
28 acceleration without magnometer:  [ 0.2736358 -0.08744284  0.4612667 ]
29 acceleration with magnometer:     [ 0.47026358 -2.37752117  0.16340941]
30 *****
31 velocity without magnometer:      [ 0.935074 -0.16917634  2.12630468]
32 velocity with magnometer:         [ 1.39086609 -7.01468068  1.46413695]
33 *****
34 location without magnometer:       [ 2.13461008 -0.34504042  5.18848774]

```

```

35 location with magnometer:      [ 2.45360254 -12.16728811  4.2801073
36 ]
37 acceleration without magnometer: [ 0.21078115 -0.07314695  0.53888497]
38 acceleration with magnometer:   [ 0.33964242 -2.34659979  0.25362898]
39 *****
40 velocity without magnometer:    [ 1.20174641 -0.47882824  2.65530114]
41 velocity with magnometer:       [ 1.79661324 -10.10131954
42   1.58828864]
43 *****
44 location without magnometer:    [ 3.44853279 -0.71032858  8.12746389]
45 location with magnometer:       [ 4.43205238 -22.67648808
46   6.16054664]
47 *****
48 acceleration without magnometer: [ 0.22118174 -0.05018395  0.50073918]
49 acceleration with magnometer:   [ 0.25730655 -2.22476348  0.24457954]
50 *****
51 velocity without magnometer:    [ 1.45657709 -0.66510888  3.18350434]
52 velocity with magnometer:       [ 2.11511748 -12.98412752
53   1.76198968]
54 *****
55 location without magnometer:    [ 5.07719701 -1.46429614 11.70659983]
56 location with magnometer:       [ 6.82642423 -36.88414884  8.2003331
57   ]
58 *****
59 acceleration without magnometer: [ 0.19707225 -0.04473991  0.38581495]
60 acceleration with magnometer:   [ 0.1566871  -2.05454365  0.16542856]
61 *****
62 velocity without magnometer:    [ 1.71391569 -0.61818085  3.71558581]
63 velocity with magnometer:       [ 2.39740111 -15.4974445
64   2.02640531]
65 *****
66 location without magnometer:    [ 7.01983304 -2.2723039  15.93238891]
67 location with magnometer:       [ 9.59354633 -54.37192878 10.5120584
68   ]

```

همانطور که دیده می‌شود این خروجی شتاب اعدادی نزدیک صفر شدند.

## جمع‌بندی

مشکلی که همچنان وجود دارد این است که به خاطر خطای اندک موجود در شتاب‌سنج، سرعت و موقعیت با کمی خطا محاسبه می‌شوند. با گذشت زمان این خطاها روی هم انباشته می‌شوند و باعث می‌شوند که موقعیت و سرعت در طی مدت کوتاهی تغییرات زیادی داشته باشند. ما در گام بعدی پروژه این سعی داریم از الگوریتم‌های دیگر که در بالا معرفی شدند استفاده کنیم و میزان خطای آنها را با یکدیگر مقایسه کنیم.