Underwater Target Tracking Based on Strong Tracking Sparse Grid Quadrature Filter

Chongyang Lv, Fei Yu*, Shu Xiao*, and Qianhui Dong

Abstract—Underwater moving object detection/tracking is critical in various applications such as exploration of natural undersea resources, acquiring of accurate scientific data to maintain regular surveillance of missions, navigation and tactical surveillance. In currently, underwater moving target is usually tracked using the traditional non-linear estimators such as Extended Kalman Filter (EKF) and unscented Kalman Filter (UKF). However, if an underwater target moves with delicate maneuver, the accuracy of the filter may decline, even diverge. In this paper, a (STSGQF) is proposed to deal with the problem. The STSGQF is obtained by introducing the Strong Tracking Filter (STF) to the Sparse Grid Quadrature Filter (SGQF). Compared with the Gauss-Hermite Quadrature Filter (GHQF), the sparse grid method is available to reduce the SGQF's computational cost significantly, with slight sacrifice of accuracy its accuracy declines slightly. Meanwhile, the STSGQF has stronger robustness than SGQF against the state change. The effectiveness of STSGQF is demonstrated by the simulation results more robust, better robustness.

Index Terms—Underwater target tracking; Gauss hermite quadrature filter; Sparse grid; Strong tracking filter

I. INTRODUCTION

The submarine plays a significant role in underwater operation, such as the exploration of marine resource and the demand of military. Detecting, classifying and tracking the underwater target around the submarine were indispensable parts of underwater defense system, which was to ensure the security of submarines [1]. So far, various types of sonar arrays and underwater sensor network had been applied, they were generally mounted on a ship, or deployed prior to the application [1]-[4]. There may be no sufficient flexibility to deal with real-time tracking missions, and the propagation delay of underwater acoustic communication may cause the decline of tracking accuracy [1]. It is important to choose a target tracking algorithm that can track the moving target expeditiously [5]-[8]. Theoretically, the underwater target tracking was a nonlinear estimate process, so the nonlinear filter algorithm is an attractive choice [9]-[11].

The Extended Kalman Filter (EKF) was widely used for state estimation in nonlinear systems [13]-[15]. However, the process of calculating the Jacobian matrix was complex and the nonlinear function must be derivable [14]. Moreover, the EKF may suffer from the problems of performance degrading and diverging in the linearization process [15]. Alcocer et.al proposed the Unscented Kalman Filter (UKF), in which the Unscented Transformation (UT) was adopted to propagate meaning and covariance information [14], [16], [17]. As compared with the EKF, the UKF did not need to calculate the Jacobian matrix and its approximation of posterior distribution can reach 3rd accuracy level [16]. However, the effect of UKF was not good in a high dimensional strong nonlinear system [14]. On the basis of Gauss Hermite Quadrature (GHQ) rule, the Gauss Hermite Quadrature Filter (GHQF) was proposed by Ito et al.[18]-[22]. In theory, the GHQF can reach any order accuracy by selecting the number of the one dimensional quadrature points, but the computational complexity of the GHQF grows exponentially with the system dimension, which was known as "curse of dimensionality" [21], [22]. To solve this problem, the Sparse Grid Quadrature Filter (SGQF) based on the Sparse Grid Quadrature (SGQ) rule was presented [19]-[22]. The SGQ rule is based on a special linear combination of lower level sensor products to extend the one dimensional point set [21]-[23]. Therefore, the number of quadrature points under the SGQ rule was obviously less than that of the GHQ rule [19]. In terms of the point selection strategy, the SGQF was more flexible and it has higher accuracy than UKF [22].

However, if the target state suddenly changes, the accuracy of the proposed filters above (EKF, UKF, GHQF and SGQF) may decline and even diverge due to model errors [14]. The adaptive algorithm is a good strategy to solve the problem. One of them, the Strong Tracking Filter (STF) was proposed by Zhou et al. to overcome the poor robustness[24], in which a fading factor was introduced to provide better state estimation by adjusting the gain matrix in real time [24]-[28]. STF had several important merits, including: 1) strong robustness against model uncertainties; 2) better real time state tracking capability, no matter whether the system reaches steady or not; and 3) moderate computational cost[23]-[25]. To improve the accuracy and simplify the computation, a filter called Strong Tracking Unscented Kalman Filter (STUKF) was develop based on the combination of UKF and STF[27]; [28]. The accuracy of SGQF was higher than UKF. In this paper, a novel point based adaptive nonlinear filter algorithm called Strong
Tracking Sparse Grid Quadrature Filter (STSGQF) was proposed. In the STSGQF, the suboptimal fading factor of STF was introduced to time update and measurement update equations of the SGQF to provide better robustness and accuracy.

The rest of the paper was organized as follows: in section II, the GHQF was briefly reviewed and the SGQ rule was introduced by a concrete example. In section III, the STF was reviewed and the STSGQF was derived. Two simulation examples were given in section IV, various kinds of filters were illustrated in weak and strong maneuver conditions, respectively. Conclusions were given in section V.

### II. SPARSE GRID QUADRATURE FILTER

The SGQF based on the SGQ rule and the Kalman Filter framework was briefly introduced.

#### A. Gauss-Hermite Quadrature Filter

Consider a nonlinear system given by

\[
\dot{x}_k = f(x_{k-1}) + w_k
\]
\[
z_k = h(x_k) + v_k
\]

where \( x_k \) is the state vector and \( z_k \) is the measurement vector; \( w_k \) and \( v_k \) are independent white Gaussian process noise and measurement noise with covariance \( Q_k \) and \( R_k \), respectively.

The univariate GHQ rule is given by

\[
\int f(x)N(x;0,1)dx \approx \sum_{i=1}^{s} \omega_i f(\chi_i)
\]

where \( f(x) \) is the integrand; \( s \) is the number of quadrature points; \( \chi_i \) and \( \omega_i \) are quadrature points and weights obtained by quadrature rule, respectively.

The univariate points and their weights can be calculated as follows.

First of all, constructing a symmetric tri-diagonal matrix \( J \) with zero diagonal elements and \( J_{i,i+1} = \sqrt{i/2} \) \((i = 1, 2, \ldots, s - 1)\). Then the j-th quadrature point \( \chi_j \) is calculated by \( \chi_j = \sqrt{2} \lambda_j \), where \( \lambda_j \) is the j-th eigenvector of \( J \). The corresponding weight \( \omega_j \) is calculated by \( \omega_j = |(v_j)_1|^2 \), where \( (v_j)_1 \) is the first element of the j-th normalized eigenvector of \( J \). Finally, the quadrature points and weights are obtained.

The selection of different number of quadrature points are given by Table I.

<table>
<thead>
<tr>
<th>Number of points</th>
<th>Point set ( G ) and weight set ( A )</th>
<th>Accuracy level</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>( G_i = {0} ) ( A_i = {1} )</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>( G_i = {-p_i, 0, p_i} = (-1.7321, 0, 1.7321) )</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>( G_i = {-p_i, -p_i, 0, p_i, p_i} = (-2.857, -1.356, 0, 1.356, 2.857) )</td>
<td>3</td>
</tr>
</tbody>
</table>

The multivariate GHQF algorithm is summarized as follows.

1) Initialization

\[
\hat{x}_0 = E[x_0] \quad \quad P_0 = \text{cov}(x_0)
\]

where \( \hat{x}_0 \) is the initial state vector; \( P_0 \) is the initial state covariance matrix.

2) Prediction

\[
P_{k|k-1} = S_{k-1}^T S_{k-1}
\]
\[
\xi_{i,k|k-1} = S_{k-1} \cdot \chi_i + \hat{x}_{k-1}
\]
\[
\hat{x}_{k|k-1} = \sum_{i=1}^{M} \omega_i f(\xi_{i,k|k-1})
\]
\[
P_{k|k-1} = \sum_{i=1}^{M} \omega_i (f(\xi_{i,k|k-1}) - \hat{x}_{k|k-1})(f(\xi_{i,k|k-1}) - \hat{x}_{k|k-1})^T + Q_{k-1}
\]

3) Update

\[
P_{z,z,k|k-1} = \sum_{i=1}^{M} \omega_i (h(\xi_{i,k|k-1}) - \hat{z}_{k|k-1})(h(\xi_{i,k|k-1}) - \hat{z}_{k|k-1})^T + R_k
\]
\[
P_{z,x,k|k-1} = \sum_{i=1}^{M} \omega_i (f(\xi_{i,k|k-1}) - \hat{x}_{k|k-1})(h(\xi_{i,k|k-1}) - \hat{z}_{k|k-1})^T
\]
\[
P_{z,x,k|k-1} = \sum_{i=1}^{M} \omega_i (f(\xi_{i,k|k-1}) - \hat{x}_{k|k-1})(h(\xi_{i,k|k-1}) - \hat{z}_{k|k-1})^T
\]
\begin{equation}
K_k = P_{zz,k|k-1} P_{zz,k|k-1}^{-1}
\end{equation}
\begin{equation}
\hat{x}_k = \hat{x}_{k|k-1} + K_k (z_k - \hat{z}_{k|k-1})
\end{equation}
\begin{equation}
P_k = P_{k|k-1} - K_k P_{zz,k|k-1} K_k^T
\end{equation}
where \( \hat{z}_{k|k-1} \) is the transformed point; \( P_{zz,k|k-1} \) and \( P_{zz,k|k-1}^{-1} \) are innovation covariance matrix and cross covariance matrix, respectively; \( \hat{z}_{k|k-1} \) is the predicted measurement; \( K_k \) is the gain matrix; \( \hat{x}_k \) and \( P_k \) are the updates of state vector and state covariance matrix, respectively.

B. Sparse grid quadrature rule

As can be seen in the Eq. (4), \( s^n \) quadrature points were needed to solve the multivariate numerical integration. The computational cost grows exponentially as dimension increases, hence the "curse of dimension"[21]. The sparse-grid method utilized a linear combination of low-level sensor products to approximate the multivariate integral [18].

The multivariate numerical integral based on sparse-grid method was shown as follows.

\begin{equation}
I = \int f(x)N(x; 0, I_n)dx \approx I_{n,L}(f)
\end{equation}
\begin{equation}
= \sum_{q=L-n}^{L-1} (-1)^L C_{n-1}^{n-1} \sum_{\Phi \in N_n^q} (I_i \otimes \cdots \otimes I_i)(f)
\end{equation}
where \( x = [x_1, x_2, \cdots, x_n]^T \in \mathbb{R}^n \): \( I_{n,L}(f) \) is an approximation to the \( n \)-dimensional integral with accuracy level \( L \), which means that \( I_{n,L}(f) \) is accurate to all the polynomials of the form \( x_1^i_1 x_2^i_2 \cdots x_n^i_n \), and satisfies the condition of \( \sum_{j=1}^{n} i_j \leq 2L - 1 \). The binomial coefficient is marked as \( C_{n-1}^{n-1} \); The sensor product is marked as \( \otimes \).

\( I_i(f) \) is the \( j \)-th univariate integral with the accuracy level of \( i_j \in \Phi \), and \( \Phi \triangleq \{i_1, i_2, \cdots, i_n\} \) is an accuracy level sequence. Accuracy level \( i_j \) means \( I_i(f) \) is accurate up to the \( (2i_j - 1) \)-th order of the univariate polynomial. \( N_n^q \) is the set of accuracy level sequences, it is defined as

\begin{align}
N_n^q &= \left\{ \Phi: \sum_{j=1}^{n} i_j = n + q \right\}, \quad q \geq 0 \\
N_n^q &= \emptyset, \quad q < 0
\end{align}
where \( q \) is a parameter of nonnegative integer and satisfies the condition of \( L - n \leq q \leq L - 1 \); \( \emptyset \) is the empty set.

The point set of the univariate integral \( I_i(f) \) is marked as \( X_i \). The univariate quadrature points in \( X_i \) are selected through the univariate GHQ rule mentioned in section A. \( x_s = [x_{s_1}, \cdots, x_{s_n}]^T \) is the \( s \)-th quadrature point of the \( n \)-dimensional vector and \( x_{s_j} \in X_i \); \( \omega_{s_j} \) is the corresponding weight in \( I_{i_j}(f) \) associated with the point \( x_{s_j} \). The set of \( n \)-dimensional sparse-grid quadrature points is given by

\begin{equation}
X_{n,L} = \bigcup_{q=L-n}^{L-1} \bigcup_{\Phi \in N_n^q} (X_i \otimes \cdots \otimes X_i)
\end{equation}
where \( \bigcup \) is denoted as the union of the quadrature point sets.

Each element in \( N_n^q \) can build a special sensor product \( X_{i_j} \), where \( i_j \in \Phi \in N_n^q \).

The SGQ rule can be regarded as that an \( n \)-dimensional integral is expanded by the univariate integral through a particular strategy of point selection. The explicit form of the Eq. (17) can be written as

\begin{equation}
I_{n,L}(f) = \sum_{q=L-n}^{L-1} \sum_{\Phi \in N_n^q} \sum_{s_1 \in X_{i_1}} \cdots \sum_{s_n \in X_{i_n}} f(x_{s_1}, \cdots, x_{s_n}) \times \left\{ (-1)^L C_{n-1}^{n-1} \prod_{j=1}^{n} \omega_{s_j} \right\}
\end{equation}
The selection strategy of the SGQ points and the corresponding weights is shown as follows.

1) The dimension \( n \) and accuracy level \( L \) of the multivariate integral are determined at first;
2) Parameter \( q \) and the set of accuracy level sequences \( N_n^q \) can be calculated by Eq. (18);
3) \( \Phi \triangleq \{i_1, i_2, \cdots, i_n\} \) is obtained, then the SGQ points like \( x_{s_j} \) are obtained;
4) Finally, calculating the corresponding weight \( \omega_{s_j} \). If the point is new, add it to the set of SGQ points \( \chi \), and calculate the corresponding weight of \( \chi_s \) as follows

\begin{equation}
W_s = (-1)^{L-q} C_{n-1}^{n-1} \prod_{j=1}^{n} \omega_{s_j}
\end{equation}
If the point already exists, make it stay and update its weight by recursion addition. To explain the SGQ rule specifically, the process of quadrature points with accuracy level \( L = 3 \) is analyzed. The value of \( q \) can be 0, 1 or 2 and the dimension is \( n \geq 3 \). The univariate quadrature points and weights of different accuracy are shown in TABLE 1.

When \( q = 0 \), \( \sum_{j=1}^{n} i_j = n \), so
As a result, the number of the SGQ points with $L = 3$ is

$$X_{n,3} = 2n^2 + 4n + 1, \ n \geq 2$$  \hspace{1cm} (22)

The corresponding weights are

$$W_3 = \begin{cases}
(1) [0,0, \ldots, 0]^T \\
(n-1)(n-2) \\
\quad - (n-1)\omega_1 + n\omega_1 + \frac{n(n-1)}{2} \omega_1^2 \\
(n-1)\omega_2 + (n-1)\omega_1\omega_2 \\
\omega_3^2 \\
\quad (2) \text{the type of } [\pm p_1, 0, \ldots, 0]^T \\
\omega_4 \\
\quad (4) \text{the type of } [\pm p_2, 0, \ldots, 0]^T \\
\omega_5 \\
\quad (5) \text{the type of } [\pm p_1, 0, \ldots, 0]^T
\end{cases}$$  \hspace{1cm} (23)

Similarly, the number of the SGQ points with $L = 2$ is

$$X_{n,2} = 2n + 1, \ n \geq 2$$  \hspace{1cm} (24)

The weights $W_2$ are

$$W_2 = \begin{cases}
(1) [0,0, \ldots, 0]^T \\
(n-1) + n\omega_1 \\
\quad (2) \text{the type of } [\pm p_1, 0, \ldots, 0]^T \\
\omega_2 \\
\quad (3) \text{the type of } [\pm p_1, \pm p_1, 0, \ldots, 0]^T \\
\omega_4 \\
\quad (4) \text{the type of } [\pm p_2, 0, \ldots, 0]^T \\
\omega_5 \\
\quad (5) \text{the type of } [\pm p_1, 0, \ldots, 0]^T
\end{cases}$$  \hspace{1cm} (25)

For example, the process of 2-dimensional point selection was shown intuitively in Fig. 1.

![Fig.1 Sparse grid points and traditional points](image)

The quadrature points $\chi$ and the weights $W$ obtained by SGQ rule were substituted in GHQF, then the SGQF was derived.

According to the literature [18], the $n$-variate integral $I_{n, L}$ in the Eq. (17) was exact for $n$-variate polynomials of the total order up to $2L - 1$. Then the SGQF with $L = 3$ was superior to the EKF and the UKF, furthermore, its computation was far less than the GHQF.
III. **STRONG TRACKING SPARSE GRID QUADRATURE FILTER**

The STF was introduced to improve the robustness of the SGQF. The STF showed significant reduction of system uncertainty to the system uncertainty like unpredictable disturbances created by external condition and model uncertainties [23].

A. **Strong Tracking Filter**

The STF can timely adjust the gain matrix with a fading factor to deal with sudden changes. So the predicted covariance matrix $P_{kk-1}$ should be changed as

$$P_{kk-1} = \lambda_k K_{kk-1} P_{kk-1} F_{kk-1}^T + Q_{kk-1}$$

where $\lambda_k$ was the fading factor. Its suboptimal solution was

$$\lambda_k = \left\{ \begin{array}{ll}
\lambda_0, & \lambda_0 \geq 1 \\
1, & \lambda_0 < 1
\end{array} \right.$$

where

$$\lambda_0 = \frac{\text{tr} [N_k]}{\text{tr} [M_k]}$$

Thus

$$K_{kk-1} = V_k - H_k Q_{kk-1} H_k^T - R_k$$

$$M_k = H_k F_{kk-1} P_{kk-1} F_{kk-1}^T H_k^T$$

and

$$F_{kk-1} = \frac{\partial f(x_{k-1})}{\partial x_{k-1}} $$

$$H_k = \frac{\partial h_k(x_k)}{\partial x_{k-1}}$$

where $\text{tr}(\bullet)$ was the trace of matrix; $V_k$ was the covariance matrix of the residual sequence, it was unknown and can be estimated by

$$V_k = \frac{\rho V_{k-1} + e_k e_k^T}{1 + \rho}$$

where $0 < \rho \leq 1$ was the forgetting factor, usually $\rho = 0.95$; $e_k$ was the residual sequence as follows

$$e_k = z_k - \hat{z}_{kk-1}$$

The revised predicted covariance matrix took the place of the original matrix, then we can get the STF.

B. **Strong Tracking Sparse Grid Quadrature Filter**

Before the fading factor was introduced, the predicted covariance matrix was marked as $P_{kk-1}$, the innovation covariance matrix was marked as $P_{zz, kk-1}$, the cross covariance matrix was marked as $P_{xz, kk-1}$, and the predicted measurement was marked as $\hat{z}_{kk-1}$ . Process noise and measurement noise were independent, so

$$F_{zz, kk-1} = E[(x_k - \hat{x}_{kk-1})(x_k - \hat{x}_{kk-1})^T]$$

$$= E[(H_k (x_k - \hat{x}_{kk-1}) + w_k - v_k) [(H_k (x_k - \hat{x}_{kk-1}) + w_k - v_k)^T]$$

$$= H_k E[(x_k - \hat{x}_{kk-1})(x_k - \hat{x}_{kk-1})^T] H_k^T + E[(w_k - v_k)(w_k - v_k)^T]$$

$$= H_k P_{\hat{x},kk-1} H_k^T + R_k$$

and

$$P_{zz, kk-1} = P_{kk-1} F_{kk-1}^T$$

then

$$H_k = [P_{zz, kk-1}]^T (P_{zz, kk-1})^{-1}$$

The Eq. (33), (34) and (35) were put into the Eq. (29), the equivalent expressions of $N_k$ and $M_k$ were given by

$$N_k = V_k - R_k - [P_{zz, kk-1}]^T (P_{zz, kk-1})^{-1} Q_{kk-1} (P_{zz, kk-1})^{-1} P_{zz, kk-1}$$

$$M_k = P_{zz, kk-1} - V_k + N_k$$

and the fading factor $\lambda_k$ can be calculated by Eq. (27)~(36).

The process of the STSGQF based on the SGQF was given as follows.

1) $P_{kk-1}$, $P_{zz, kk-1}$, $P_{xz, kk-1}$ and $\hat{z}_{kk-1}$ can be calculated by Eq. (5)~(13);

2) The fading factor $\lambda_k$ is calculated by Eq. (27)~(36);

3) The fading factor $\lambda_k$ is introduced to the original predicted covariance matrix $P_{kk-1}$, and it can be changed as

$$P_{kk-1} = \lambda_k \sum_{i=1}^{M} w_i (f(x_{kk-1}) - \hat{x}_{kk-1})(f(x_{kk-1}) - \hat{x}_{kk-1})^T + Q_{kk-1}$$

4) According to the Eq. (9)~(13), the new innovation covariance matrix $P_{zz, kk-1}$, the new cross covariance matrix $P_{xz, kk-1}$ and the new predicted measurement $\hat{z}_{kk-1}$ with fading factor are obtained;

5) Update the filter with $P_{zz, kk-1}$, $P_{xz, kk-1}$ and $\hat{z}_{kk-1}$ by the Eq. (14)~(16).

IV. **RESULTS AND DISCUSSION**

For both weak and strong maneuver, we compared the performances of different filters under the two conditions.

A. **Weak maneuver**

In Cartesian coordinate, we assumed that the underwater target is in motion with constant velocity of $(10, 5, 0) \text{ m/s}$ at the initial position $(200, 100, -100) \text{ m}$ in the directions of $x$, $y$ and $z$; the period $T$ is 0.01s, and tracking the target 300 steps; 100 Monte-Carlo simulations were carried out.

When choosing CA model, the state vector is

$$x_k = \begin{bmatrix} x & y & z & \dot{x} & \dot{y} & \dot{z} \end{bmatrix}^T$$

State transition matrix is

$$F = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

(Advance online publication: 10 February 2018)
Initial state covariance matrix is
\[
P = \text{diag}(\begin{bmatrix} x_0^2 & y_0^2 & z_0^2 & \dot{x}_0^2 & \dot{y}_0^2 & \dot{z}_0^2 \end{bmatrix})
\]

Process noise covariance matrix is
\[
Q = \begin{bmatrix} Q_1 & 0 & 0 \\ 0 & Q_1 & 0 \\ 0 & 0 & Q_1 \end{bmatrix}
\]

\[
Q_1 = \begin{bmatrix} T^5 / 20 & T^4 / 8 & T^3 / 6 \\ T^4 / 8 & T^3 / 3 & T^2 / 2 \\ T^3 / 6 & T^2 / 2 & T \end{bmatrix}
\]

Measurement noise covariance matrix is
\[
R = \text{diag}(\{\rho \theta \phi\})
\]

Where \( \rho \) was the error of radial distance, and \( \theta \) was the pitch and azimuth’s errors.

Five filters were compared, GHQFs with \( L = 2 \) and \( L = 3 \), SGQFs with \( L = 2 \) and \( L = 3 \), and UKF with \( \kappa = 0 \) were marked as GHQF2, GHQF3, SGQF2, SGQF3 and UKF(0), respectively. The Root Mean Square Error (RMSE) of position marked as RMSE-s and the RMSE of velocity marked as RMSE-v in Fig. 2 and Fig. 3.

![Fig. 2 Position Root Mean Square Error](image)

According to the result in Fig. 2, it is clear that the GHQF3 performed the best and the tracking accuracy of the SGQF2 was the worst. SGQFs performed worse than GHQFs with the same accuracy level. However, the SGQF3 was superior to the GHQF2 and the UKF (0). In Fig. 3, the relationship of five filters’ RMSE-v was similar to RMSE-s.

The performance of algorithm was evaluated not only by considering its result but also its efficiency. The performance of filters was presented in Table Ⅱ.

<table>
<thead>
<tr>
<th>Filter</th>
<th>Time/s</th>
</tr>
</thead>
<tbody>
<tr>
<td>GHQF2</td>
<td>1371.311</td>
</tr>
<tr>
<td>GHQF3</td>
<td>28295.863</td>
</tr>
<tr>
<td>SGQF2</td>
<td>38.515</td>
</tr>
<tr>
<td>SGQF3</td>
<td>197.390</td>
</tr>
<tr>
<td>UKF(0)</td>
<td>25.498</td>
</tr>
</tbody>
</table>

Result in Table Ⅱ showed that the GHQFs run much longer than the SGQFs with the same accuracy level, especially the GHQF3 has run nearly 8 hours. Although the GHQF3 performs the best, it is the worst in terms of efficiency. The SGQF2 shortens about 97% running time contrasted with the GHQF2, only need more than half minute, yet its effect is not expected. The SGQF3 runs for more than 3 minutes, and shortens about 99% time contrasted with the GHQF3, furthermore, its tracking accuracy is slightly lower than the GHQF3 and better than the other 3 filters. The UKF (0) spends less than half minute on processing, but its accuracy is not high enough. Considering comprehensively, the SGQF3 is the best among the 5 filters.

### B. Strong maneuver

In Cartesian coordinate, we assumed that the underwater target is in motion with velocity of \((10, -5, -1)\)m/s at the initial position \((200, 100, -100)\)m in the directions of \(x\), \(y\) and \(z\), respectively; tracking the target 500 steps. Other parameters...
and initial variables were presented in section A. Changes of acceleration of the target were shown in Table III.

<table>
<thead>
<tr>
<th>step (i)</th>
<th>(x)-acceleration /m/s(^2)</th>
<th>(y)-acceleration /m/s(^2)</th>
<th>(z)-acceleration /m/s(^2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-100</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>100-200</td>
<td>-10</td>
<td>20</td>
<td>-5</td>
</tr>
<tr>
<td>200-300</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>300-400</td>
<td>-10</td>
<td>-20</td>
<td>15</td>
</tr>
<tr>
<td>400-500</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

In this test, the performances of the STF, the STUKF, the STSGQF3 and the SGQF3 were compared. Trajectory of the target was presented in Fig. 4. The RMSE-\(s\) and the RMSE-\(v\) are given by Fig. 5 and Fig. 6.

In order to verify the effectiveness which reducing the computation load of STSGQF, the Monte Carlo method was used. We sampled 100,000 random numbers between 0 and 1. These values were combined with STF, STUKF, STSGQF and SGQF, respectively. The average run time of STF, STUKF, STSGQF\(_3\) and SGQF\(_3\) were 272.356s, 380.377s, 583.798s and 710.254s respectively as seen in Table IV.

It can be seen from Fig. 4 that the underwater target makes sharp turns. The results of tracking were shown in Fig. 5 and Fig. 6, the accuracy of strong tracking algorithms was higher than the SGQF3, especially when the state changes, and they converged more quickly than the SGQF3. Among the strong tracking algorithms, the STSGQF3 was the best overall.

Considering the filters’ running time, the computational cost of the STF was the least, the STSGQF3 cost about 10 minutes. Compared with the SGQF3, the running time of the STSGQF3 increased, but the STSGQF3 was improved against the system uncertainties with increasing the running time.

<table>
<thead>
<tr>
<th>Filter</th>
<th>Time/s</th>
</tr>
</thead>
<tbody>
<tr>
<td>STF</td>
<td>272.356</td>
</tr>
<tr>
<td>STUKF</td>
<td>380.377</td>
</tr>
<tr>
<td>STSGQF(_3)</td>
<td>583.798</td>
</tr>
<tr>
<td>SGQF(_3)</td>
<td>710.254</td>
</tr>
</tbody>
</table>

V. CONCLUSIONS

In this paper, the Strong Tracking Sparse Grid Quadrature Filter (STSGQF) is proposed. The STF is introduced to timely adjust the gain matrix, and the SGQ rule is used to select quadrature points flexibly and moderately. Simulation examples of weak and strong maneuver are provided for evaluating the performances of SGQF and STSGQF, respectively. We conclude that the SGQF can reduce the computational cost significantly compared with GHQF. After the STF is integrated into SGQF, the robustness of STSGQF is significantly improved despite the increase of running time.
Finally, in the strong maneuver case, we prove that the STSGQF can take tracking quickly and efficiently.

REFERENCES


