THE SQUARE-ROOT UNSCENTED INFORMATION FILTER FOR STATE ESTIMATION AND SENSOR FUSION

Guoliang Liu, Florentin Wörgötter and Irene Markelić
III Physikalisches Institut - Biophysik, University of Göttingen, Göttingen, Germany
{liu,worgott,irene}@physik3.gwdg.de

Keywords: Nonlinear estimation, sensor fusion, unscented information filter, square-root unscented information filter.

Abstract: This paper presents a new recursive Bayesian estimation method, which is the square-root unscented information filter (SRUIF). The unscented information filter (UIF) has been introduced recently for nonlinear system estimation and sensor fusion. In the UIF framework, a number of sigma points are sampled from the probability distribution of the prior state by the unscented transform and then propagated through the nonlinear dynamic function and measurement function. The new state is estimated from the propagated sigma points. In this way, the UIF can achieve higher estimation accuracies and faster convergence rates than the extended information filter (EIF). As the extension of the original UIF, we propose to use the square-root of the covariance in the SRUIF instead of the full covariance in the UIF for estimation. The new SRUIF has better numerical properties than the original UIF, e.g., improved numerical accuracy, double order precision and preservation of symmetry.

1 INTRODUCTION

Recently, the information filter (IF), which is the dual of the Kalman filter (KF) (Anderson and Moore, 1979), has attracted much attention for multiple sensor fusion (Wang et al., 2010; Liu et al., 2011). Both the IF and the KF represent distributions of random state variables with Gaussians. However, in contrast to moment parametrization as done in the KF, the IF uses an information matrix and an information vector to represent the Gaussians. This difference in parameterization makes the IF superior to the KF concerning multiple sensor fusion, as computations are simpler and no prior information of the system state is required (Lee, 2008).

In the case of nonlinear estimation problems, an extended version of the IF can be obtained using the first order term of the Taylor series expansions of the nonlinear functions, i.e., the dynamic and measurement functions of the system, which is called extended information filter (EIF). This approximation can introduce large errors when the system model is highly nonlinear, and the higher order terms of Taylor series are important (Van der Merwe and Wan, 2001). To address this issue, the unscented information filter (UIF) has been proposed by Kim *et al.* (Kim et al., 2008) and Lee (Lee, 2008). Kim developed the UIF by using minimum mean square error estimation. In contrast, Lee's UIF algorithm is derived by embedding statistical linear error propagation into the EIF architecture. Although their methods are different, results are essentially identical (Lee, 2008; Kim et al., 2008). The UIF uses a number of deterministic sigma points to capture the true information matrix and the information vector, which can be accurate up to the second order of any nonlinearity. These sigma points are generated by unscented transform in the UIF. As shown in (Lee, 2008), the UIF is superior to the EIF not only in terms of estimation accuracy but also concerning the convergence speed for nonlinear estimation and multiple sensor fusion. The other way to generate the sigma points is the Stirling's interpolation, which has similar performance to the unscented transform, but with fewer predefined parameters (Liu et al., 2011).

In this paper, we propose the square-root extension of the UIF. In the unscented transform, a square-root of the prior covariance has to be calculated to generate sigma points. This step is computationally expensive and requires that the covariance matrix to be positive definite. To save computational cost and increase numerical robustness, a square-root form of the covariance can be directly taken and updated in the algorithm. The square-root forms achieve better numerical characteristics than the regular ones, e.g., improved numerical accuracy, double order precision and preservation of symmetry (Arasaratnam and Haykin, 2009). In the literature, Potter (Potter and Stern., 1963) introduced the first square-root filter which was used in the Apollo manned mission. Since then, many square-root extensions of conventional filters have been introduced. Recently, Van der Merwe (Van der Merwe and Wan, 2001) has proposed square-root forms of sigma-point Kalman filters which have better numerical stability and less computational cost. Here we employ a similar idea, and introduce the square-root unscented information filter (SRUIF) for solving nonlinear state estimation and sensor fusion problems.

This paper is organized as follows: First, we first show the basic UIF algorithm for nonlinear estimation and multiple sensor fusion in Section 2, and then the square-root UIF is proposed in Section 3. Simulation results of target tracking are presented and discussed in Section 4. Finally, the work is concluded in Section 5.

2 UNSCENTED INFORMATION FILTER

The *Unscented Information filter* (UIF) employs the unscented transform to generate sigma points, which can be further used to estimate the mean and covariance of the system state. The UIF algorithm is summarized in Algorithm 1, where $\gamma = \sqrt{(\lambda + L)}$ is the composite scaling parameter, $\lambda = \alpha^2(L+\kappa) - L$, α and κ are scaling parameters that determine how far the sigma points spread from the mean value (Van der Merwe, 2004), L is the dimension of the state, R_v and R_n are process noise covariance and observation noise covariance respectively, $w_i^{(m)}$ and $w_i^{(c)}$ are weights calculated by $w_0^m = \frac{\lambda}{L+\lambda}$, $w_0^c = \frac{\lambda}{L+\lambda} + (1 - \alpha^2 + \beta)$, $w_i^m = w_i^c = \frac{1}{2(L+\lambda)}$, $i = 1, \cdots, 2L$.

2.1 Global Information Fusion

In case of multiple sensors N, where the measurement noises between the sensors are uncorrelated, the measurement update for information fusion is simply expressed as a linear combination of the local information contribution terms:

$$y_k = y_k^- + \sum_{i=1}^{N} \phi_{i,k} \tag{1}$$

$$Y_k = Y_k^- + \sum_{i=1}^{N} \Phi_{i,k}, \tag{2}$$

where $\phi_{i,k}$ and $\Phi_{i,k}$ are defined in (14) and (15) respectively.

Algorithm 1 UIF for State Estimation

```
• Initialization:
  $\hat{x}_0 = E(x_0),\ P_{x_0} = E\left((x_0 - \hat{x}_0)(x_0 - \hat{x}_0)^T\right).$
• For $k = 1, \cdots, \infty$:
  1. Generate sigma points for prediction:
     $\hat{x}^{a_v}_{k-1} = \begin{bmatrix}\hat{x}_{k-1}\\ \bar{v}\end{bmatrix},\quad P^{a_v}_{k-1} = \begin{bmatrix}P_{x_{k-1}} & 0\\ 0 & R_v\end{bmatrix}$   (3)
     $\mathcal{X}^{a_v}_{k-1} = \left[\hat{x}^{a_v}_{k-1} \quad \hat{x}^{a_v}_{k-1} + \gamma\sqrt{P^{a_v}_{k-1}} \quad \hat{x}^{a_v}_{k-1} - \gamma\sqrt{P^{a_v}_{k-1}}\right]$   (4)
  2. Prediction equations:
     $\mathcal{X}^x_{k|k-1} = f(\mathcal{X}^x_{k-1}, \mathcal{X}^v_{k-1}, u_{k-1})$   (5)
     $\hat{x}^-_k = \sum_{i=0}^{2L} w_i^{(m)} \mathcal{X}^x_{i,k|k-1}$   (6)
     $P^-_{x_k} = \sum_{i=0}^{2L} w_i^{(c)} \left(\mathcal{X}^x_{i,k|k-1} - \hat{x}^-_k\right)\left(\mathcal{X}^x_{i,k|k-1} - \hat{x}^-_k\right)^T$   (7)
     $Y^-_k = (P^-_{x_k})^{-1}$   (8)
     $\hat{y}^-_k = Y^-_k \hat{x}^-_k$   (9)
  3. Generate sigma points for measurement update:
     $\mathcal{X}_{k|k-1} = \left[\hat{x}^-_k \quad \hat{x}^-_k + \gamma\sqrt{P^-_{x_k}} \quad \hat{x}^-_k - \gamma\sqrt{P^-_{x_k}}\right]$   (10)
  4. Measurement update equations:
     $\mathcal{Z}_{k|k-1} = h\left(\mathcal{X}_{k|k-1}\right)$   (11)
     $\hat{z}^-_k = \sum_{i=0}^{2L} w_i^{(m)} \mathcal{Z}_{i,k|k-1}$   (12)
     $P_{x_k z_k} = \sum_{i=0}^{2L} w_i^{(c)} \left[\mathcal{X}_{i,k|k-1} - \hat{x}^-_k\right]\left[\mathcal{Z}_{i,k|k-1} - z^-_k\right]^T$   (13)
     $\phi_k = Y^-_k P_{x_k z_k} R_n^{-1} \left[z_k - z^-_k + (P_{x_k z_k})^T y^-_k\right]$   (14)
     $\Phi_k = Y^-_k P_{x_k z_k} R_n^{-1} (P_{x_k z_k})^T (Y^-_k)^T$   (15)
     $y_k = y^-_k + \phi_k$   (16)
     $Y_k = Y^-_k + \Phi_k$   (17)
```

3 SQUARE-ROOT UIF

The square-root UIF benefits from three powerful matrix factorization techniques: *QR decomposition*, *Cholesky factor updating* and *efficient least squares*. In the following, we will use *qr*, *chol*, *cholupdate* and '\' (backslash) to refer to the *QR decomposi-*

Algorithm 2 Square-Root UIF for State Estimation

- Initialization:
  $\hat{x}_0 = E(x_0)$, $S_v = \sqrt{R_v}$ and $S_n = \sqrt{R_n}$, $S_{x0} = chol\left\{E\left((x_0-\hat{x}_0)(x_0-\hat{x}_0)^T\right)\right\}$.
- For $k = 1, \cdots, \infty$:
  1. Generate sigma points for prediction:

$$\hat{x}^{a_v}_{k-1} = \begin{bmatrix}\hat{x}_{k-1}\\ \bar{v}\end{bmatrix}, \quad S^{a_v}_{k-1} = \begin{bmatrix}S_{x_{k-1}} & 0\\ 0 & S_v\end{bmatrix} \tag{18}$$

$$\mathcal{X}^{a_v}_{k-1} = \left[\hat{x}^{a_v}_{k-1} \quad \hat{x}^{a_v}_{k-1}+\gamma S^{a_v}_{k-1} \quad \hat{x}^{a_v}_{k-1}-\gamma S^{a_v}_{k-1}\right] \tag{19}$$

  2. Prediction equations:

$$\mathcal{X}^x_{k|k-1} = f(\mathcal{X}^x_{k-1}, \mathcal{X}^v_{k-1}, u_{k-1}) \tag{20}$$

$$\hat{x}^-_k = \sum_{i=0}^{2L} w_i^{(m)} \mathcal{X}^x_{i,k|k-1} \tag{21}$$

$$S^-_{x_k} = qr\left\{\sqrt{w_1^{(c)}}\left(\mathcal{X}^x_{1:2L,k|k-1} - \hat{x}^-_k\right)\right\} \tag{22}$$

$$C = \sqrt{w_0^{(c)}}\left(\mathcal{X}^x_0 - \hat{x}^-_k\right) \tag{23}$$

$$S^-_{x_k} = cholupdate\left\{S^-_{x_k}, C, sign\{w_0^{(c)}\}\right\} \tag{24}$$

$$\hat{y}^-_k = \left(S^-_{x_k}\right)^T \backslash \left(S^-_{x_k}\backslash \hat{x}^-_k\right) \tag{25}$$

$$S^-_{y_k} = qr\left\{S^-_{x_k}\backslash I\right\} \tag{26}$$

  3. Generate sigma points for measurement update:

$$\mathcal{X}_{k|k-1} = \left[\hat{x}^-_k \quad \hat{x}^-_k+\gamma S^-_{x_k} \quad \hat{x}^-_k-\gamma S^-_{x_k}\right] \tag{27}$$

  4. Measurement update equations:

$$\mathcal{Z}_{k|k-1} = h\left(\mathcal{X}_{k|k-1}\right) \tag{28}$$

$$\hat{z}^-_k = \sum_{i=0}^{2L} w_i^{(m)} \mathcal{Z}_{i,k|k-1} \tag{29}$$

$$P_{x_k z_k} = \sum_{i=0}^{2L} w_i^{(c)} [\mathcal{X}_{i,k|k-1} - \hat{x}^-_k][\mathcal{Z}_{i,k|k-1} - z^-_k]^T \tag{30}$$

$$U = \left(S^-_{x_k}\right)^T \backslash \left(S^-_{x_k}\backslash P_{x_k z_k}\right) / S^T_n \tag{31}$$

$$y_k = \hat{y}^-_k + U/S_n(z_k - \hat{z}^-_k + P^T_{x_k z_k}\hat{y}^-_k) \tag{32}$$

$$S_{y_k} = cholupdate\{S^-_{y_k}, U, +1\} \tag{33}$$

tion, *Cholesky decomposition*, *Cholesky factor updating* and *efficient least squares* respectively [1].

- *QR decomposition*. In the UIF, the square-root of the covariance matrix S is derived by *Cholesky decomposition* on P: $S = chol(P)^T$ where S is a lower triangular matrix and fulfills $P = SS^T$. If we know $P = AA^T$, the square-root factor S can be directly calculated from A by *QR decomposition*: $S = qr(A)^T$. If the matrix $A \in \mathbb{R}^{L\times N}$, then the computational complexity of a *QR decomposition* is $O(NL^2)$.
- *Cholesky factor updating*. If the original update of the covariance matrix is $P \pm uu^T$ and S is the Cholesky factor, then the rank 1 update of S is $S = cholupdate(S, u, \pm)$ where u is the update vector. If u is a matrix, we can update each column of u one by one in a loop. For each column vector, the computational complexity is $O(L^2)$. This procedure can alternatively be implemented as $S = qr([S \pm u]^T)$ using *QR decomposition* without the loop updates.
- *Efficient least squares*. The least squares solution for the linear equation $Px = b$ can be solved efficiently using forward and back substitution if the Cholesky factor S is known and satisfies $P = SS^T$. For example, we can solve it by $x = S^T\backslash(S\backslash b)$ where '\' is the backslash. This operation only requires computational complexity $O(L^2)$.

The whole process is shown in Algorithm 2, which looks similar to the general UIF algorithm introduced in Section 2, except that the *Cholesky factor* S_{x_k} is used instead of the covariance P_k. The square-root UIF also comprises two steps, the first is the prediction and the second is the measurement update. For each step, a number of sigma points are generated using *unscented transform* in (19) and (27). However, the square root of covariance S_{x_k} is directly used to calculate the sigma points without the *Cholesky factorization*.

In the prediction step, the *Cholesky factor* S_{x_k} is updated using *QR decomposition* on the weighted sigma points. This step replaces the P_k update in (7) and has complexity $O(L^3)$. The information vector $\hat{y}^-_k = (P^-_k)^{-1}\hat{x}^-_{x_k} = \left(S^-_{x_k}\right)^T \backslash \left(S^-_{x_k}\backslash\hat{x}^-_k\right)$ is derived using *efficient least squares* in (25). Because $\hat{S}^-_{x_k}$ is a square and triangular matrix, we can directly use back-substitution for solving $\hat{y}^-_k$ without the need for matrix inversion. The back substitution only requires $O(L^2)$. Next is the calculation of the square-root information matrix $S^-_{y_k}$ in (26). This step requires a *QR decomposition* since $S^-_{y_k}$ is a upper triangular matrix, and meets $\left(S^-_{y_k}\right)^T S^-_{y_k} = \left(S^-_{x_k}\right)^{-T}\left(S^-_{x_k}\right)^{-1}$. As $S^-_{x_k}$ is a lower triangular matrix, *QR decomposition* is used to

[1] The abbreviations *qr*, *chol*, *cholupdate* and '\' (backslash) are in accordance with the function names for *QR decomposition*, *Cholesky decomposition*, *Cholesky factor updating* and *efficient least squares* in Matlab.

solve the *Cholesky factor* $S^-_{y_k}$ of the information matrix $Y_{k|k-1}$. To avoid the inversion, here we use *efficient least squares* to solve $\left(S^-_{x_k}\right)^{-1}$ as $S^-_{x_k}\backslash I$, where I is an identity matrix.

In the measurement update step, the updated information vector y_k is derived by *efficient least squares* in (32). If the observation dimension is M, the updated square-root information matrix S_{y_k} is calculated in (33) by applying an M-sequential Cholesky update to $S^-_{y_k}$. The columns of matrix U are update vectors. This requires $O(L^2M)$ and replaces the measurement update of Y_k in (17).

3.1 Square-Root UIF for Multiple Sensor Fusion

In the case where information from multiple sensors is available, i.e., $N > 1$, we can fuse this using the square-root UIF. For the i_{th} sensor, the information contribution for the information vector is

$$\phi_{i,k} = U/S_n^T(z_k - \hat{z}_k^- + P^T_{x_k z_k}\hat{y}_k^-) \tag{34}$$

where U is defined in (31). The information contribution for the square-root information matrix is

$$S_{i,\phi_k} = U. \tag{35}$$

The final estimated result is derived by:

$$y_k = \hat{y}_k^- + \sum_{i=0}^{N} \phi_{i,k} \tag{36}$$

$$S_{y_k} = cholupdate\{S^-_{y_k}, [S_{1,\phi_k} \quad S_{2,\phi_k} \quad \cdots \quad S_{N,\phi_k}], +1\}. \tag{37}$$

4 EXPERIMENTS

In this section, we consider a nonlinear bearing-only tracking (BOT) problem using the UIF and SRUIF and compare their performances. The bearing-only tracking problem has become an important benchmark for different probability inference methods (Bar-Shalom et al., 2001; Sadhu et al., 2004). Hartikainen and Särkkä (Hartikainen and Särkkä, 2008) have developed a toolbox which includes the comparison between the UKF, the EKF, and their smoothers by solving the BOT problem with static sensors.

Here we use the same system model as in (Hartikainen and Särkkä, 2008). A moving target object is tracked by two static angular sensors as shown in Fig. 1. The discrete time update of the dynamic object on time step k is

$$\mathbf{x}_k = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \mathbf{x}_{k-1} + \mathbf{v}_{k-1} \tag{38}$$

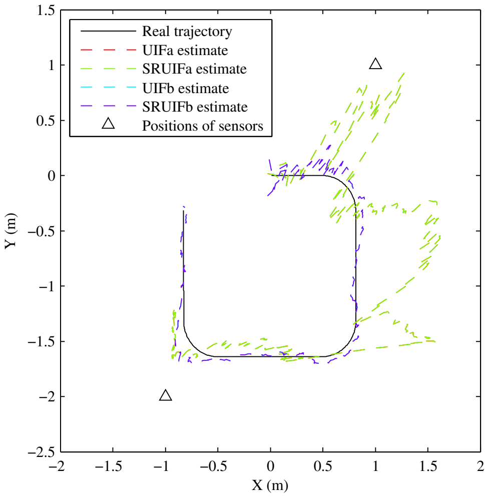

Figure 1: The ground truth and estimated trajectories for the bearing-only tracking. In the figure, the estimated trajectories from UIFa and SRUIFa are overlapped, and the estimated trajectories from UIFb and SRUIFb are overlapped too.

where the system state is $\mathbf{x}_k = (x_k, y_k, \dot{x}_k, \dot{y}_k)^T$, which includes the target position (x_k, y_k) and velocity $(\dot{x}_k, \dot{y}_k)$. Δt is the time interval between time step k and $k-1$, which is set to $\Delta t = 0.01$ in the simulation. $\mathbf{v}_{k-1}$ is Gaussian noise with zero mean and the covariance is

$$\mathbf{R}_v = \begin{bmatrix} \frac{1}{3}\Delta t^3 & 0 & \frac{1}{2}\Delta t^2 & 0 \\ 0 & \frac{1}{3}\Delta t^3 & 0 & \frac{1}{2}\Delta t^2 \\ \frac{1}{2}\Delta t^2 & 0 & \Delta t & 0 \\ 0 & \frac{1}{2}\Delta t^2 & 0 & \Delta t \end{bmatrix} \xi \tag{39}$$

where ξ is the spectral density of the noise (Hartikainen and Särkkä, 2008) and set to $\xi = 0.1$ in our experiment. The target is tracked by sensors located at (x_s, y_s), where $s = 1,2$ in the case of two sensors. The measurement model of the s_{th} sensor is defined as

$$\theta_s = tan^{-1}\left(\frac{y_k - y_s}{x_k - x_s}\right) + e_{\theta,s} \tag{40}$$

where $e_{\theta,s} \sim \mathcal{N}(0, R_{n,s})$ is the measurement noise of the s_{th} sensor. The sensors are located at $(x_1, y_1) = (-1, -2)$ and $(x_2, y_2) = (1, 1)$, and their measurement noise variances are $R_{n,1} = R_{n,2} = 0.05^2$. The initial prior state $\hat{\mathbf{x}}_0$ and the covariance $\hat{\mathbf{P}}_0$ are given by:

$$\hat{\mathbf{x}}_0 = [0,\ 0,\ 1,\ 0]^T \tag{41}$$

$$\hat{\mathbf{P}}_0 = diag(0.1, 0.1, 10, 10), \tag{42}$$

where *diag* means the diagonal matrix.

The estimated results from different filters are summarized in Table 1. We also show the estimated trajectories in the Fig. 1. It can be seen that the UIF and SRUIF have equal accuracy for nonlinear estimation and sensor fusion, but the SRUIF is slightly faster than the UIF. Van der Merwe (Van der Merwe and Wan, 2001) shown that the square-root UKF can be 20% faster than the UKF, but in our case the SRUIF only achieves 3% faster than the UIF.

Table 1: Means (E) and standard deviations (STD) of RMSE values of the position and average run time (T) in 100 Monte Carlo runs of the bearing-only tracking. UIFa and SRUIFa use one sensor, whereas UIFb and SRUIFb use two sensors.

Method	E	STD	T (s)
UIFa	0.6794	0.1725	0.5102
SRUIFa	0.6794	0.1725	0.4944
UIFb	0.1145	0.0283	0.8154
SRUIFb	0.1145	0.0283	0.7763

5 CONCLUSION

In this paper, we present the square-root UIF for nonlinear estimation and sensor fusion. It has the same computational complexity as the original UIF, i.e., $O(L^3)$. However, the SRUIF has better numerical properties, such as the improved numerical accuracy, double order precision and preservation of symmetry. In addition since the square-root of the covariance matrix is directly available, the SRUIF can save computational costs in the step of sigma-points calculation. The experimental results show that the SRUIF runs slightly faster than the original UIF. In the future, we plan to investigate their performances with different sensor network architectures (Lee, 2008), and further improve the estimation accuracies, e.g., by combining the proposed filters with the adaptive consensus algorithm (Casbeer and Beard, 2009).

REFERENCES

Anderson, B. D. O. and Moore, J. B. (1979). *Optimal Filtering*. Eaglewood Cliffs, NJ: Prentice-Hall.

Arasaratnam, I. and Haykin, S. (2009). Cubature kalman filters. *Automatic Control, IEEE Transactions on*, 54(6):1254 –1269.

Bar-Shalom, Y., Li, X.-R., and Kirubarajan, T. (2001). *Estimation with Applications to Tracking and Navigation*. Wiley Interscience.

Casbeer, D. and Beard, R. (2009). Distributed information filtering using consensus filters. In *American Control Conference, 2009. ACC '09.*, pages 1882 –1887.

Hartikainen, J. and Särkkä, S. (2008). Optimal filtering with kalman filters and smoothers a manual for matlab toolbox ekf/ukf.

Kim, Y., Lee, J., Do, H., Kim, B., Tanikawa, T., Ohba, K., Lee, G., and Yun, S. (2008). Unscented information filtering method for reducing multiple sensor registration error. In *Multisensor Fusion and Integration for Intelligent Systems, IEEE International Conference on*, pages 326 –331.

Lee, D.-J. (2008). Nonlinear estimation and multiple sensor fusion using unscented information filtering. *Signal Processing Letters, IEEE*, 15:861 –864.

Liu, G., Wörgötter, F., and Markelić, I. (2011). Nonlinear estimation using central difference information filter. In *IEEE Workshop on Statistical Signal Processing*, pages 593–596.

Potter, J. and Stern., R. (1963). Statistical filtering of space navigation measurements. In *In AIAA Guidance Contr. Conference*.

Sadhu, S., Srinivasan, M., Mondal, S., and Ghoshal, T. (2004). Bearing only tracking using square root sigma point kalman filter. In *India Annual Conference, 2004. Proceedings of the IEEE INDICON 2004. First*, pages 66 – 69.

Van der Merwe, R. (2004). *Sigma-Point Kalman Filters for Probabilistic Inference in Dynamic State-Space Models*. PhD thesis, OGI School of Science & Engineering, Oregon Health & Science University,.

Van der Merwe, R. and Wan, E. (2001). The square-root unscented kalman filter for state and parameter-estimation. In *Acoustics, Speech, and Signal Processing, 2001. Proceedings. (ICASSP '01). 2001 IEEE International Conference on*, volume 6, pages 3461 – 3464 vol.6.

Wang, L., Zhang, Q., Zhu, H., and Shen, L. (2010). Adaptive consensus fusion estimation for msn with communication delays and switching network topologies. In *Decision and Control (CDC), 2010 49th IEEE Conference on*, pages 2087 –2092.