

Multiple Object Tracking Intern Report

Hasan Atakan Bedel
Department of Electrical and Electronics Engineering
Middle East Technical University

CONTENTS

I	Introduction	2
II	Problem Definition and Approach	2
III	Objectives	2
IV	Project Structure	2
V	IMM-UKF-JPDAF	2
V-A	IMM	2
V-A1	Mixing	3
V-A2	Mode Prediction Updates	3
V-A3	Mode Measurement Updates	3
V-A4	New Mode Probabilities	3
V-A5	Output Estimate Calculation	3
V-B	UKF	3
V-B1	Unscented Transform	3
V-B2	Prediction	4
V-B3	Second Unscented Transform	4
V-B4	Calculate Update Parameters	4
V-B5	Update	4
V-C	JPDAF	4
V-C1	Validation Matrix	4
V-C2	Association Events	4
V-C3	Joint Association Probabilities	5
V-D	PDAF	6
V-E	Putting It All Together	6
V-E1	Mixings	6
V-E2	Predictions	6
V-E3	Joint Data Association Probabilities Calculation	6
V-E4	Updates	6
V-E5	New Mode Probabilities Calculation	6
V-E6	Final Output Estimations	6
V-F	Testing of the Algorithms	6
VI	Trackers	6
VI-A	Single Object Single Model Tracker	6
VI-B	Single Object Multiple Model Tracker	6
VI-C	Multiple Object Single Model Tracker	6
VI-D	Multiple Object Multiple Model Tracker	6
VII	Trackers in Action	6
VIII	The Need For Optimization	6
VIII-1	Optimization	6
IX	Performance Assesment	6
X	Fullfilment of the Objectives	6
XI	Conclusion	6

References	6
-------------------	----------

LIST OF FIGURES

1	Project Structure	2
---	-----------------------------	---

LIST OF TABLES

I	Notation Table	2
II	IMM Notation Table	3
III	UKF Notation Table	3
IV	JPDAF Notation Table	5

Multiple Object Tracking Intern Report

I. INTRODUCTION

Multiple object tracking is an essential part of the perception pipeline. In order to solve this problem many methods have been developed. My internship subject was to approach this problem using classical methods which are proposed in the master thesis provided to me. To be more precise used methods are: IMM(Interaction Multiple Model) for modeling object trajectories with different kinematic models, UKF(Unscented Kalman Filter) for nonlinear filtering of the obtained measurements and JPDAF(Joint Probability Data Association Filter) for a sophisticated data association method. I have implemented all of them in python and prepared mock data to assess their results.

II. PROBLEM DEFINITION AND APPROACH

The objective of multiple object tracking is to link and filter object detections across the sampled timeline. This way each object's trajectory and state estimation can be obtained. The difference between single object tracking and multiple object tracking is, later needs to associate object detections with existing tracks so that their states can be estimated correctly based on these associated detections.

The supplied master thesis uses UKF(Unscented Kalman Filter) to estimate the object states. Kalman filter needs a prediction model to predict the next state. In the thesis multiple kinetic models are used; Constant Velocity(CV), Constant Turn-Rate Velocity(CTRV) Model and Random Motion Model(RM). CTRV has nonlinear nature, hence the need for UKF. In order to use these different models to give only one unified estimation IMM is used.

In order to track multiple objects an data association method is required. JPDAF is used for this purpose. JPDAF is a soft data association method, meaning it does not associate the detections with existing tracks one to one. It associates all detections to all tracks based on their calculated joint association probabilities.

III. OBJECTIVES

*Explain why you had to implement the algorithms yourself
*Talk about other possible methods to solve multi object tracking problem

To be honest, no clear objectives were provided in the beginning of my internship, only the master thesis. After have read the thesis and searched internet for some context I have determined the objectives myself, which were seem to be approved by my supervisor Dr. Berker Logoglu. Here is the list.

- Implementation of IMM, UKF, JPDAF algorithms
- Development of an test environment for the algorithms using mock data
- Test of the algorithms on Nuscenes dataset

IV. PROJECT STRUCTURE

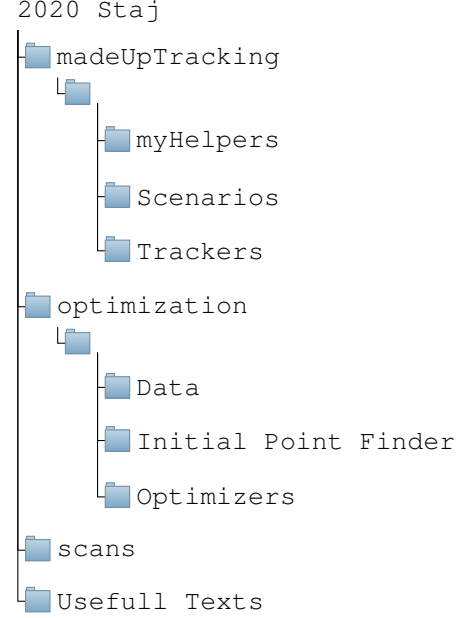


Fig. 1: Project Structure

V. IMM-UKF-JPDAF

In this section, I will give details about the algorithms and their implementation details with some insights I have gained.

Indices	
K	Number of ...
T	Number of ...
Parameters	
c	something

TABLE I: Notation Table

A. IMM

IMM is used to integrate multiple kinematic models to obtain fused estimate. All used models have their own weights(mode probabilities), which determine their influence on the fused estimate. Mode probabilities are updated at each measurement time step, based on if one mode(kinematic model) were more successfull to estimate at the previous time step then the other modes.

Notation	
r_k	The mode at time step k
z_k	Measured state at k
x_k	State at k
\hat{x}_k	State mean at k
\hat{z}_k	State measured mean at k
\hat{x}_k^i	State mean at k according to mode i
$\hat{x}_k^{[i]}$	Mixed state mean at k for mode i
S_k	State innovation matrix at k
Z_k	The measurement at time k
μ_k^i	Mode probability at k

TABLE II: IMM Notation Table

1) *Mixing*: At each step all the models have to mix their states with each other. This is the interaction step where IMM(Interaction Multiple Model)'s 'I' take it's name. This step helps models to stay in some bound(not formal) without diverging from each other completely.

$$\mu_{k-1|k-1}^{ji} \triangleq P(r_{k-1} = j | r_k = i, z_{0:k-1}) \quad (1)$$

$$\mu_{k-1|k-1}^{ji} = \frac{\pi_{ji} \mu_{k-1}^j}{\sum_{l=1}^{N_r} \pi_{li} \mu_{k-1}^l} \quad (2)$$

$$\hat{x}_{k-1|k-1}^{[i]} = \sum_{j=1}^{N_r} \mu_{k-1|k-1}^{ji} \hat{x}_{k-1|k-1}^j \quad (3)$$

$$P_{k-1|k-1}^i = \sum_{j=1}^{N_r} \mu_{k-1|k-1}^{ji} [P_{k-1|k-1}^j + (\hat{x}_{k-1|k-1}^j - \hat{x}_{k-1|k-1}^{[i]})(\hat{x}_{k-1|k-1}^j - \hat{x}_{k-1|k-1}^{[i]})^T] \quad (4)$$

2) *Mode Prediction Updates*: In this part each model independently predicts the next state, which are state mean and state covariance.

3) *Mode Measurement Updates*: Again in this part each model independently updates their state using the measurement(s) given.

4) *New Mode Probabilities*: Here each model's mode probability is calculated. New mode probabilities are dependent on how good each model were able estimate the new state using markov chains.

$$\mu_k^i = \frac{N(Z_k; \hat{z}_{k|k-1}^i, S_k^i) \sum_{j=1}^{N_r} \pi_{ji} \mu_{k-1}^j}{\sum_{l=1}^{N_r} N(Z_k; \hat{z}_{k|k-1}^l, S_k^l) \sum_{j=1}^{N_r} \pi_{jl}} \quad (5)$$

5) *Output Estimate Calculation*: Finally the one unified estimate is calculated based on the new calculated mode probabilities. Each model can contribute to the unified estimate proportional to their mode probability. Note that this stage does not affect any part of the algorithm, it is only used for output.

$$\hat{x}_{k|k} = \sum_{i=1}^{N_r} \mu_k^i \hat{x}_{k|k}^i \quad (6)$$

$$P_{k|k} = \sum_{i=1}^{N_r} \mu_k^i [P_{k|k}^i + (\hat{x}_{k|k}^i - \hat{x}_{k|k})(\hat{x}_{k|k}^i - \hat{x}_{k|k})^T] \quad (7)$$

B. UKF

It is used to estimate state for non-linear prediction models. Best intuitive and formal explanation is in it's original paper[REFERENCE HERE]

Notation	
$\hat{x}_{k-1 k-1}^i$	i'th sigma point obtained from k-1'th step

TABLE III: UKF Notation Table

1) *Unscented Transform*: First sigma points are calculated. It is like sampling near the state mean, but in a systematic and proven way. Note that $(\sqrt{(L+\lambda)P_{k-1|k-1}})_i$ is the i'th column or row of the matrix inside the parentheses. [REFERENCE HERE] But one must stick to one choice, row or column.

$$\begin{aligned} \hat{x}_{k-1|k-1}^0 &= \hat{x}_{k-1|k-1} \\ \hat{x}_{k-1|k-1}^i &= \hat{x}_{k-1|k-1} + (\sqrt{(L+\lambda)P_{k-1|k-1}})_i \\ &\quad i \in 1, \dots, L \\ \hat{x}_{k-1|k-1}^i &= \hat{x}_{k-1|k-1} - (\sqrt{(L+\lambda)P_{k-1|k-1}})_{i-L} \\ &\quad i \in L+1, \dots, 2L \end{aligned} \quad (8)$$

2) *Prediction*: At this part each sigma points are passed through the prediction function. After that, each points are used to get the predicted state mean and covariance. The used parameter values in my implementation for the parameters presented here($\alpha, \kappa, \beta...$) can be found in the original paper. [REFERENCE HERE]

$$\begin{aligned}
 \hat{x}_{k|k-1}^i &= f(\hat{x}_{k-1|k-1}^i), \quad i \in 0, \dots, 2L \\
 W_s^0 &= \frac{\lambda}{L + \lambda}, \quad W_c^0 = \frac{\lambda}{L + \lambda} + (1 - \alpha^2 + \beta) \\
 W_s^i &= W_c^i = \frac{1}{2(L + \lambda)}, \quad \lambda = \alpha^2(L + \kappa) - L \\
 \hat{x}_{k|k-1} &= \sum_{i=0}^{2L} W_s^i \hat{x}_{k|k-1}^i \\
 P_{k|k-1} &= \sum_{i=0}^{2L} W_c^i [\hat{x}_{k|k-1}^i - \hat{x}_{k|k-1}] [\hat{x}_{k|k-1}^i - \hat{x}_{k|k-1}]^T + Q
 \end{aligned} \tag{9}$$

3) *Second Unscented Transform*: In prediction state the state covariance matrix is calculated and now it is time to find the new sigma points with respect to this new predicted state mean and covariance. Why do we need these sigma points? Reason is to make measurements on these sigma points. If the measurement function is linear than this part and next part is not necessary. Update parameters(S_k, K_k) can be calculated using the classical kalman filter equations. This is the case with my project.

$$\begin{aligned}
 \hat{x}_{k|k-1}^{*0} &= \hat{x}_{k|k-1} \\
 \hat{x}_{k|k-1}^{*i} &= \hat{x}_{k|k-1} + \begin{pmatrix} \sqrt{(L + \lambda) P_{k|k-1}} \end{pmatrix}_i \\
 &\quad i \in 1, \dots, L \\
 \hat{x}_{k|k-1}^{*i} &= \hat{x}_{k|k-1} - \begin{pmatrix} \sqrt{(L + \lambda) P_{k|k-1}} \end{pmatrix}_{i-L} \\
 &\quad i \in L + 1, \dots, 2L
 \end{aligned} \tag{10}$$

4) *Calculate Update Parameters*: If the measurement function is linear, which is the case for my situation, than this part can be replaced with classical kalman filter update parameter calculations. In my implementation I have coded both of them to see the difference. Note that I have not seen any formal explanation to do this but it made sense and indeed they give the exact same result.

$$\begin{aligned}
 \hat{z}_k^i &= h(\hat{x}_{k|k-1}^i), \quad i \in 0, \dots, 2L \\
 \hat{z}_{k|k-1} &= \sum_{i=0}^{2L} W_c^i [\hat{x}_{k|k-1}^{*i} - \hat{x}_{k|k-1}] [\hat{x}_{k|k-1}^{*i} - \hat{x}_{k|k-1}]^T \\
 P_{z_{k|k-1}, z_{k|k-1}} &= \sum_{i=0}^{2L} W_c^i [\hat{z}_{k|k-1}^i - \hat{z}_{k|k-1}] [\hat{z}_{k|k-1}^i - \hat{z}_{k|k-1}]^T \\
 S_k &= P_{z_{k|k-1}, z_{k|k-1}} + R \\
 P_{x_{k|k-1}, z_{k|k-1}} &= \sum_{i=0}^{2L} [\hat{x}_{k|k-1}^i - \hat{x}_{k|k-1}] [\hat{z}_{k|k-1}^i - \hat{z}_{k|k-1}]^T \\
 K_k &= P_{x_{k|k-1}, z_{k|k-1}} S^{-1}
 \end{aligned} \tag{11}$$

5) *Update*: These are usual kalman filter update equations.

$$\begin{aligned}
 \hat{x}_k &= \hat{x}_{k|k-1} + K_k (Z_k - \hat{z}_{k|k-1}) \\
 P_k &= P_{k|k-1} - K_k S K_k^T
 \end{aligned} \tag{12}$$

C. JPDAF

JPDAF is used for soft data association. It assumes that the number of existing tracks is a known, meaning it does not have power to initiate new tracks from the given measurements. It calculates the possibility of associations between each track and each measurement. This data association probability is the output of the JPDAF. Later these probabilities are passed to PDAF along with the measurements to update the each tracker. JPDAF is clearly explained in [REFERENCE], but implementation part was not clear.

1) *Validation Matrix*: "Validation Matrix" is presented here [REFERENCE HERE]. It is a binary matrix with m_k number of rows and $N_r + 1$ number of columns. The elements of the matrix, lets say $b_{i,j}$, represent if the i 'th measurement is close to j 'th track. If it is close, that element is 1 otherwise 0. Whether a measurement is close to a track is decided based on the mahalanobis distance between the track's measured state mean and the measurement, using the innovation matrix as the covariance. The rationale can be find in [REFERENCE]. Note that $j = 0$ represents the possibility of that measurement is associated with no track. To include this possibility for all measurements the first column of the validation matrix is initiated as 1.

2) *Association Events*: An association event is nothing but just one of the possibilites to associate the existing tracks and the incoming measurements. In my implementation I have written an exhaustive algorithm($O(m_k^{N_r+1})$) to try find the possible association events using the validation matrix. If the validation matrix is dense, meaning the all measurements are close to all tracks, and if there are vast number of tracks and measurements the performance of the algorithms is very bad. I measured the run time of my implementation with $m_k = 8$ and $N_r = 8$ and filling the validation matrix with all ones, as 41.797 seconds, which is very bad. On the other hand with

$m_k = 5$ and $N_r = 5$ and filling the validation matrix with all ones, as 0.036 seconds. Note that a more sparse validation matrix really helps to time performance, but downside it can not be sparsity can not be guaranteed. [CODE REFERENCE]

3) *Joint Association Probabilities*: All possible events were generated and stored. Now it is time to calculate their probability. Now here I give the formal definition of an event.

$$\theta = \bigcap_{j=1}^m \theta_{j,t_j} \quad (13)$$

where θ_{j,t_j} is the event that measurement j is associated with track t_j . Track t_j is the associated track with measurement j. Now here are the two ways of calculating the probability of an event.

Parametric calculation:

$$P(\theta|Z^k) = \frac{1}{c} \prod_{j=1}^{m_k} \{\lambda^{-1} f_{t_j}[z_j(k)]\}^{\tau_j} \prod_{t=1}^{N_r} (P_D)^{\delta_t} (1 - P_D)^{1-\delta_t} \quad (14)$$

Nonparametric calculation:

$$P(\theta|Z^k) = \frac{1}{c'} \phi! \prod_{j=1}^{m_k} \{V f_{t_j}[z_j(k)]\}^{\tau_j} \prod_{t=1}^{N_r} (P_D)^{\delta_t} (1 - P_D)^{1-\delta_t} \quad (15)$$

$$f_{t_j}[z_j(k)] = N[Z_k^j, \hat{z}_{k|k-1}^{t_j}, S_k^{t_j}] \quad (16)$$

The derivations can be found in [REFERENCE].

Notation	
V	Volume of the surveillance region
λ	Spatial density of false measurements
P_D	Detection probability of the detector
δ_t	A binary number showing whether target t is associated with a measurement in that event
τ_j	A binary number showing whether measurement j is associated with a target in that event
t_j	It is the index of the track which measurement j is associated with in that event
ϕ	The number of false alarms in that event

TABLE IV: JPDAF Notation Table

The implementation of surveillance region was not clear in [BYL95]. One solution could be to take the volume of the track with highest volume (page 211 [BYL95]), or maybe summing the max volumes of all tracks could be a solution. I haven't implemented the non-parametric version since it is ambiguous.

Note that the calculation of event probabilities are not normalized (there is a $\frac{1}{c}$ constant in the event probability calculation). That is why we need to normalize the event probabilities now.

$$P(\theta|Z^k) = \frac{P(\theta|Z^k)}{\sum_{\theta' \in \text{all events}} P(\theta'|Z^k)} \quad (17)$$

After normalization it is time to calculate the data association probabilities needed for the PDAF part.

$$\beta_k^{jt} \triangleq P(\theta_{jt}|Z_k) = \sum_{\theta: \theta_{jt} \in \theta} P(\theta|Z_k) \quad (18)$$

where θ_{jt} is the event that measurement j is associated with target t.

Equation(18) is for $j \in 1, \dots, m_k$. For $j=0$, meaning no measurement is associated with the track t , here is the calculation.

$$\beta_k^{0t} = 1 - \sum_{j=1}^{m_k} \beta_k^{jt} \quad (19)$$

D. PDAF

PDAF is about updating a track with multiple measurements each of which has a association probability with that track. In fact, one of the measurements is "no measurement available for track" which also has a probability. [BYL95 page 130] Here

$$\begin{aligned} v_k^j &= Z_k^j - \hat{z}_{k|k-1} \\ v_k &= \sum_{j=1}^{m_k} \beta_k^{jt} v_k^j \end{aligned} \quad (20)$$

E. Putting It All Together

Bla bla

- 1) Mixings:
- 2) Predictions:
- 3) Joint Data Association Probabilities Calculation:
- 4) Updates:
- 5) New Mode Probabilities Calculation:
- 6) Final Output Estimations:

F. Testing of the Algorithms

The main difference between this section and the one in your report *Do not include your findings in this section.*

VI. TRACKERS

In order to test algorithms independently, I have created 4 different tracking classes. First I have implemented Single Object Single Model Tracker and then moved to the others until have finished the final goal Multiple Object Multiple Model Tracker.

A. Single Object Single Model Tracker

It is used to track single object with single model. The measurement to be fed to the tracker in each sampling time is expected to exist and be one. The algorithm of the interest to be validated is *UKF*.

*Reference to code location *How it is used *Rationale about the validation of the target algorithms

B. Single Object Multiple Model Tracker

*What it does and target algorithms to be validated *Reference to code location *How it is used *Rationale about the validation of the target algorithms

C. Multiple Object Single Model Tracker

*What it does and target algorithms to be validated *Reference to code location *How it is used *Rationale about the validation of the target algorithms

D. Multiple Object Multiple Model Tracker

*What it does and target algorithms to be validated *Reference to code location *How it is used *Rationale about the validation of the target algorithms

VII. TRACKERS IN ACTION

The main difference between this section and the one in your report *Do not include your findings in this section.*

VIII. THE NEED FOR OPTIMIZATION

Use the subsection command with caution—you probably won't need it at, but I'm including it this an example.

1) *Optimization*: Use the subsection command with caution—you probably won't need it at, but I'm including it this an example.

IX. PERFORMANCE ASSESMENT

Use the subsection command with caution—you probably won't need it at, but I'm including it this an example.

X. FULLFILMENT OF THE OBJECTIVES

Use the subsection command with caution—you probably won't need it at, but I'm including it this an example.

XI. CONCLUSION

Use the subsection command with caution—you probably won't need it at, but I'm including it this an example.

REFERENCES

- [1] H. Kopka and P. W. Daly, *A Guide to L^AT_EX*, 3rd ed. Harlow, England: Addison-Wesley, 1999.
- [2] D. Horowitz, *End of Time*. New York, NY, USA: Encounter Books, 2005. [E-book] Available: ebrary, <http://site.ebrary.com/lib/sait/Doc?id=10080005>. Accessed on: Oct. 8, 2008.
- [3] D. Castelvetti, "Nanoparticles Conspire with Free Radicals" *Science News*, vol.174, no. 6, p. 9, September 13, 2008. [Full Text]. Available: Proquest, <http://proquest.umi.com/pqdweb?index=52&did=1557231641&SrchMode=1&sid=3&Fmt=3&VInst=PROD&VType=PQD&RQT=309&VName=PQD&TS=1229451226&clientId=533>. Accessed on: Aug. 3, 2014.
- [4] J. Lach, "SBFS: Steganography based file system," in *Proceedings of the 2008 1st International Conference on Information Technology, IT 2008, 19-21 May 2008, Gdansk, Poland*. Available: IEEE Xplore, <http://www.ieee.org>. [Accessed: 10 Sept. 2010].
- [5] "A 'layman's' explanation of Ultra Narrow Band technology," Oct. 3, 2003. [Online]. Available: <http://www.vmsk.org/Layman.pdf>. [Accessed: Dec. 3, 2003].