

Implementation of Storing Data Algorithm for Tracking Targets using LABVIEW®

Hazem Kamel

Radar Department, Military Technical College
Cairo, Egypt

M. N. Eltoukhy

Radar Department, Military Technical College
Cairo, Egypt

Abstract— Human operators of any radar or aviation system can only update a maximum of six tracks about once per two seconds and only for short time intervals [1]. Therefore, researchers do their best to implement various tracking algorithms; moreover, storing the data of tracked targets has a great role in these systems such as calculating the maneuverability of targets, plotting graphs, retrieve upon request of the user. In this paper, a proposed algorithm for storing data of a tracked target is shown. Monte Carlo simulation showed that the ability of the algorithm to store every point of the target of interest. In particular, the algorithm is capable of storing X position, Y position, target ID, date, and time for the target of interest.

Keywords— Tracking, LABVIEW, database, access

I. INTRODUCTION

Storing the data of tracked targets is required in many civilian applications such as intelligent transportation system, air traffic control and surveillance. Therefore, researchers have concerned about this issue during the past years [1] in order to eliminate the usage of pencil and paper to store the date of the tracked targets. Surveillance systems are using one or more sensors together with computer subsystems to interpret the environment such as sonar, infrared (IR), and radar sensor. Reports measurement form diverse sources from field of view (FOV) containing one or more targets of interest once track initiated its data begin to store.

Since most surveillance systems must track multiple targets. Multi Target Tracking (MTT) is the most important tracking application Fig.1 [2] shows the basic element of typically MTT system. Gating purpose is to reduce computational expense by ignoring from consideration measurements which are thither from the estimated measurement position. Data association is a mean to select the candidate measurement from all the measurement located in the gate such as NN [3], PDA [4].

The measurements which correlate to a given track is processed by a filter to update the track parameter for these tracks that didn't receive correlating observations, the previous predicted estimates are treated as the filtered estimates. Then, the predictions are made to the time when the next data scan is to be received [5]

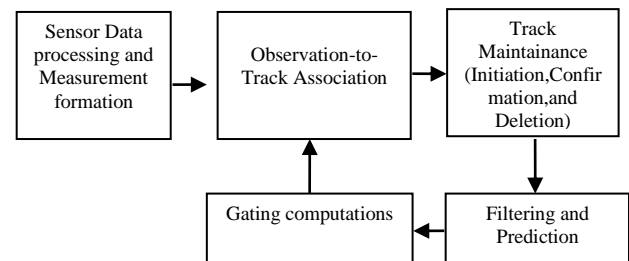


Fig.1. Basic element of MTT system [2]

In section 2 we will present flowchart of proposed multi target tracking with storing data, then section3 will show implementation of storing data algorithm using LABVIEW and section 4 the performance and evaluation, the final results for the algorithm in section 5.

II. FLOWCHART OF PROPOSED MULTI TARGET TRACKING ALGORITHM WITH STORING DATA USING LABVIEW

The flowchart of a proposed multi target tracking algorithm [8] is shown in Fig with additional part of storing algorithm. At the beginning, selected scenario is loaded and the initiation of a tentative track with quality set by 1, in the next scan according to the gate equation the gate is formed around the target.

Then, after another scan the data association part will select the suitable measurement to update the target track, and if no measurement falls inside the gate, the track will update by the predicted measurement from the previous scan and the track quality will decrease by 1. The prediction was performed using PTF.

The proposed storing algorithm which consists of a bank of storing algorithm each one of them is communicate with the proposed tracking algorithm in order to store the data of its target, the program will connect to data base access program through a udl file that created for linking LABVIEW with access, it will be discussed in next section, the data of the target which are position in x, position in y, target ID, date, and time will be stored for each target separately in access tables.

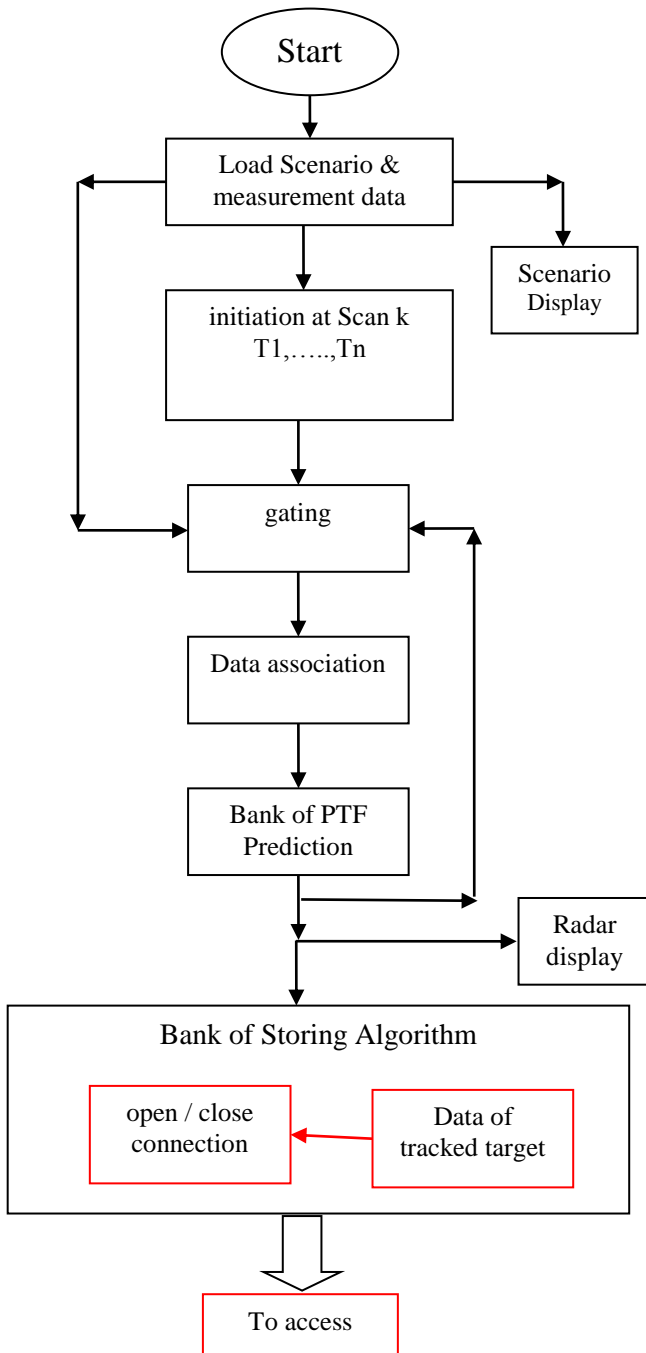


Fig.2. the flowchart of proposed multi target tracking algorithm with storing data

III. PROPOSED ALGORITHM FOR STORING DATA USING LABVIEW

The algorithm consists of two steps the first one is to make a connection between LABVIEW and access, for the second one is to implement the block diagram of the algorithm in LABVIEW and connect it with access via the path of the connection

A. Connecting LABVIEW and ACCESS

To store data of targets you have to make connection between LabVIEW and access (data base) and this is done by the following steps [6]:

- 1- Create new access (.mdb) from right click, new Microsoft access application.



Fig.3. choose access program

- 2- Create the table and the fields which are wanted.
- 3- From control panel choose administrative tools double click and choose Data Source (ODBC).

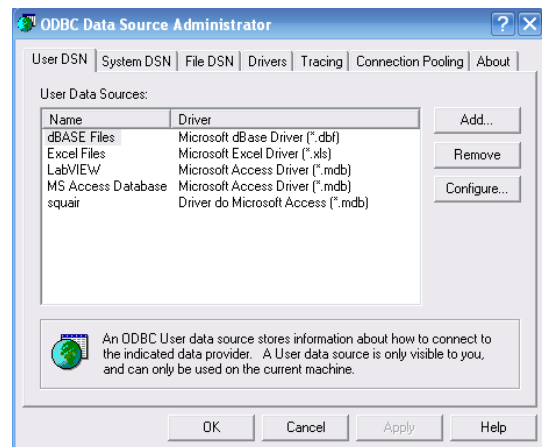


Fig.4. ODBC source

- 4- Select add from the panel and choose Driver do Microsoft Access [*.mdb].

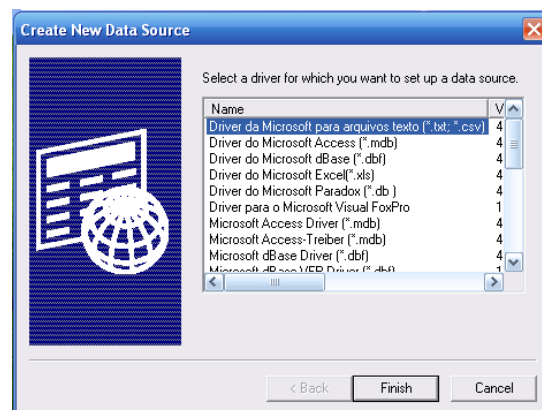


Fig.5. create new data source

- 5- Types the Data source name and describe it, then select the pass which gets the access file created before

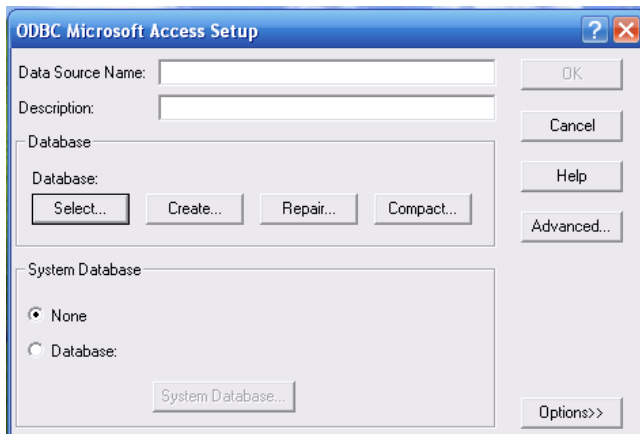


Fig.6. setup data base

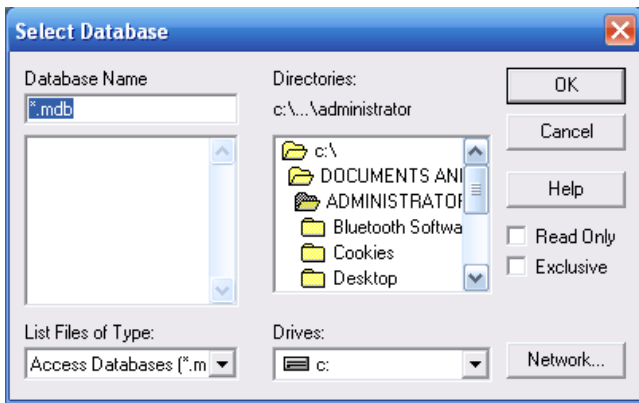


Fig.7. select data base

- 6- Create new text file then choose save as (choose the file name as u wish), and change the extension of the file to UDL file which LabVIEW program can deal with it.

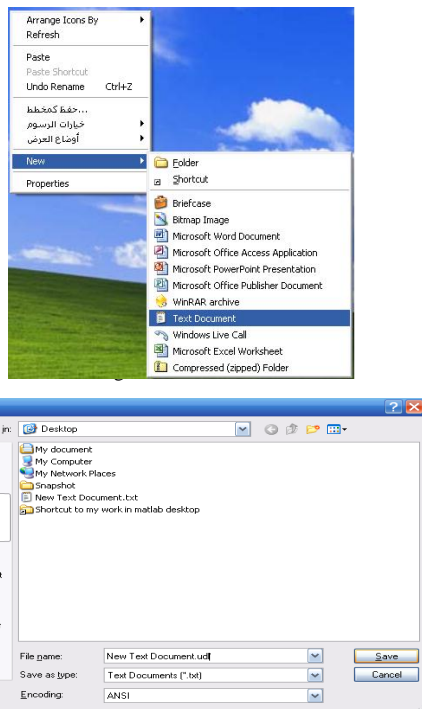


Fig.9. changing extension to dot UDL

- 7- An icon dot UDL founded, then double click and choose data source name which has created before.



Fig.10. icon dot udl

- 8- Click on test connection to be sure that the connection is succeeded.

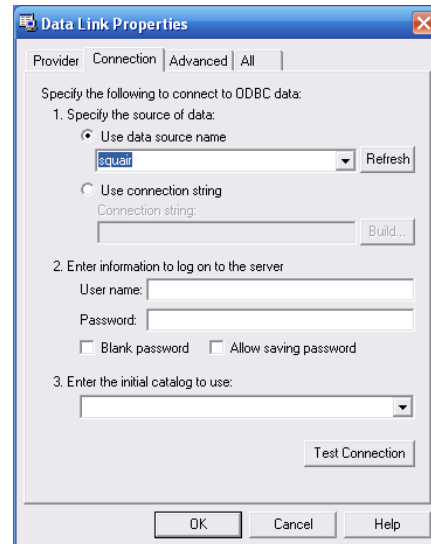


Fig.11. choose data source of data base

B. Implementation of storing data algorithm on LABVIEW

The block diagram of proposed algorithm is implemented in LABVIEW.

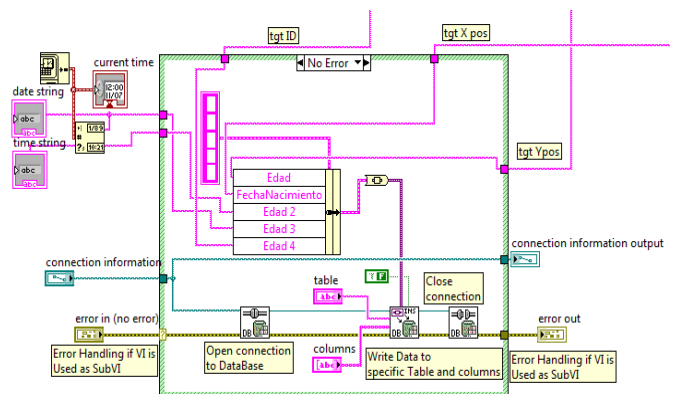


Fig.12. block diagram of insert data into data base

From the block diagram of multi target tracking in LABVIEW which implemented in [8], the storing algorithm will be inserted to the first tracking target block diagram in the tracking algorithm, in order to store the data of the first tracked target, and the required data to be stored are connected as input data to the storing algorithm.

By combining the input data as position in X, position in Y, target ID, time, and date in a cluster then connected it to the module write data to specific table and column which need to be connected with open connection to data base, close connection and connection information in a sequence as shown in Fig.12. Previously, entering the pass of the created data base from the user interface Fig 13, in addition, entering the title of each column and the title of the table by the user needs. This algorithm used to insert data of single target into data base table [9] and can be repeated according to the number of the tracking targets required depending on the processor speed.

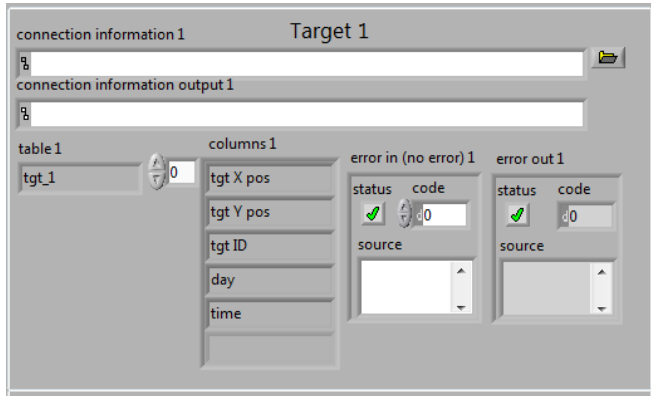


Fig.13. the front panel of data base

The errors tables in Fig 13 showed that if there is error in data base entering date or not. This appeared in messages to describe the error which was happened.

IV. PREFORMANCE ANALUSIS AND EVALUATION

In order to examine the storing algorithm, in this section we insert the storing algorithm in multi target tracking system [7] implemented in LABVIEW [8], and two different scenarios for moving targets applied, the first one for non-maneuver target, while the other one for maneuver target..

A. Non- maneuvering target

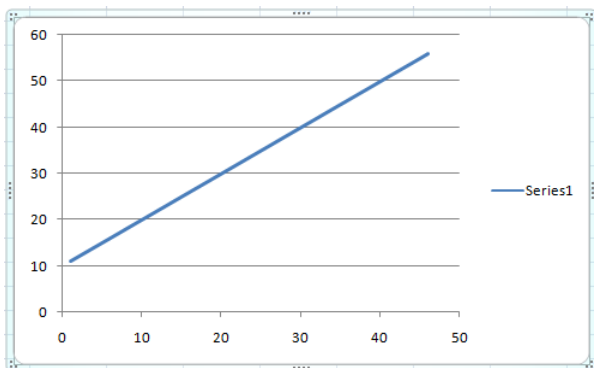


Fig.14. non-maneuver target scenario

For a linear target scenario, the storing algorithm saved the data of each measurement in the data base table without missing any one of them, moreover, the target ID, date , and time also saved as shown in Fig 15

item no.	tgt X pos	tgt Y pos	tgt ID	day	time
34	12	11	1	10/14/2016	11:17:10 AM
35	14	13	1	10/14/2016	11:17:13 AM
36	16	15	1	10/14/2016	11:17:16 AM
37	18	17	1	10/14/2016	11:17:19 AM
38	20	19	1	10/14/2016	11:17:22 AM
39	22	21	1	10/14/2016	11:17:25 AM
40	24	23	1	10/14/2016	11:17:28 AM
41	26	25	1	10/14/2016	11:17:31 AM
42	28	27	1	10/14/2016	11:17:34 AM
43	30	29	1	10/14/2016	11:17:37 AM
44	32	31	1	10/14/2016	11:17:40 AM
45	34	33	1	10/14/2016	11:17:43 AM
46	36	35	1	10/14/2016	11:17:46 AM
47	38	37	1	10/14/2016	11:17:49 AM
48	40	39	1	10/14/2016	11:17:52 AM
49	42	41	1	10/14/2016	11:17:55 AM
50	44	43	1	10/14/2016	11:17:58 AM
51	46	45	1	10/14/2016	11:18:01 AM
52	48	47	1	10/14/2016	11:18:04 AM
53	50	49	1	10/14/2016	11:18:07 AM
54	52	51	1	10/14/2016	11:18:10 AM
55	54	53	1	10/14/2016	11:18:13 AM
56	56	55	1	10/14/2016	11:18:16 AM

Fig.15. table of data for non-maneuverable target

B. Maneuvering target

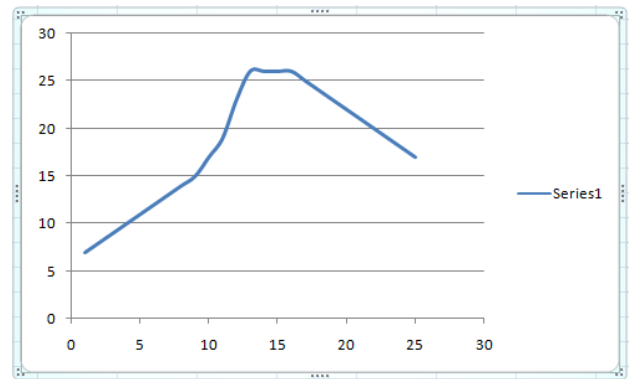


Fig.16. maneuver target scenario

For non linear target scenario, as seen in Fig 17 the storing algorithm not only, has an excellent performance in storing the data of every measurement in the data base table without ignoring any measurement of them, but also the program successfully store the target ID, date and time of each measurement stored.

item no.	tgt X pos	tgt Y pos	tgt ID	day	time
3	8	8	2	10/14/2016	11:30:48 AM
4	9	9	2	10/14/2016	11:30:51 AM
5	10	10	2	10/14/2016	11:30:54 AM
6	11	11	2	10/14/2016	11:30:57 AM
7	12	12	2	10/14/2016	11:31:00 AM
8	13	13	2	10/14/2016	11:31:03 AM
9	14	14	2	10/14/2016	11:31:06 AM
10	15	15	2	10/14/2016	11:31:09 AM
11	16	17	2	10/14/2016	11:31:12 AM
12	17	19	2	10/14/2016	11:31:15 AM
13	18	23	2	10/14/2016	11:31:18 AM
14	19	26	2	10/14/2016	11:31:21 AM
15	20	26	2	10/14/2016	11:31:24 AM
16	21	26	2	10/14/2016	11:31:27 AM
17	22	26	2	10/14/2016	11:31:30 AM
18	23	25	2	10/14/2016	11:31:33 AM
19	24	24	2	10/14/2016	11:31:36 AM
20	25	23	2	10/14/2016	11:31:39 AM
21	26	22	2	10/14/2016	11:31:42 AM
22	27	21	2	10/14/2016	11:31:45 AM
23	28	20	2	10/14/2016	11:31:48 AM
24	29	19	2	10/14/2016	11:31:51 AM
25	30	18	2	10/14/2016	11:31:54 AM

Fig.17. table of data for maneuverable target

V. CONCLUSION

We showed that the proposed algorithm for storing data for the target of interest is derived. The proposed algorithm performance is analyzed and tested for different scenarios. A graphical user interface was created. The number of targets to be stored can increase by repeating the block diagram. This algorithm can easily download on an embedded radar system as it was created by LABVIEW®.

REFERENCES

- [1] Blackman" Multi Hypothesis tracking for muti target tracking " IEEE system magazine vol 19, NO 1, PP.5-18,jan,(2004)
- [2] Y. Bar-Shalom and T. Fortmann, "Tracking and Data Association", Academic Press,(1988)
- [3] Rickard Karlsson, Fredrik Gustafsson, "Monte Carlo data association for multiple target tracking," Target Tracking: Algorithms and Applications, vol.1, 2001.
- [4] Y. Bar-Shalom, T. Kirubarajan and X. Lina "Probabilistic Data Association Techniques for Target Tracking with Applications to Sonar, Radar and EO Sensors" ,IEEE System Magazine,(2003)
- [5] Rickard Karlsson, "Simulation Based Methods for Target Tracking", Linkoping Studies in Science and Technology, Thesis No. 930, Sweden,(2002)
- [6] Database Connectivity Toolkit User Manual, LABVIEW, 371525A-01,June 2008
- [7] M. Nabil, H. Kamal, and M. Hassan, "Comparison Between Kalman Filter and PHD Filter in Multi-target Tracking," 8th International Conference on Electrical Engineering, MTC, Cairo, 2012
- [8] M. Nabil, H. Kamal, and M. Hassan" Implementation of a Proposed Multiple Target Tracking Algorithm using LabVIEW", International Journal of Engineering Research & Technology, Vol. 4 Issue 09, September-2015
- [9] M. Nabil, "Performance Improvement of Track-While-Scan System in Dense Air Traffic Control," M.Sc., MTC, Cairo, 2013