Post Boost Track Processing Using Conventional DBMS Software

Aniruddha Basu

*Defense Research & Development Laboratory, Hyderabad*

Email: [basu\_chinu@yahoo.com](mailto:basu_chinu@yahoo.com)

*Abstract*

***The design of Air Defence (AD), Command Control system is very challenging and has been treated with basic methodologies. The present design is associated with unstructured and uncorrelated data with huge lines of code*. *Hence an attempt was made for a better simplified (Database Management System) DBMS software data access methodology, on which futuristic command control system can rely. A typical Command Control Computer communication & Intelligence (C4I) subsystem algorithm was undertaken for prototype development on DBMS data access methodology. Development involved existing algorithm in C++ implemented code of post boost tracking was designed in Oracle database using Solid State drive. The purpose of this development was to find the advantage using online fast processing with a new upcoming methodology using basic Oracle database software. Used kinematic parameters generated from dynamic processing in Oracle database for determination of target Flight vehicle (FV) and decoy during separation from post boost tracks (PBTs). Earlier DBMS data access was never thought of for track processing application due to its slow disk access involving more time for transaction processing which has now being improved with the use Solid state drive apart from dynamic usage of new DBMS methodology.* *This paper presents a brief experimentation on the database (DB) methodology.***

**Keywords:** Cluster, Flight Vehicle**,** Decoy, BMD, RDBMS, DBMS, SQL, real-time, embedded

1. INTRODUCTION

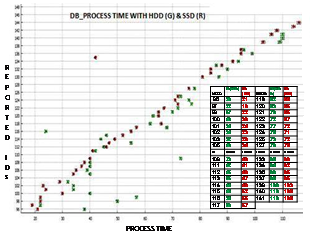
The primary objective of this paper is to explore track processing capability with RDBMS Oracle Data Base software. The goal is to reduce the complexity of application development and make it more simplified resulting from the handling of data by DBMS in real time.

This will help to realise futuristic heterogeneous command control architecture with better interoperability solution. To solve major challenge in data standardization is to make standards general enough to apply to all services and still meet the requirements of the individual specific services, proposed architecture being industry standard professional skills and manpower can be easily available than present one.

Present experiment was made on Linux environment on which Oracle 10.2 DB was installed with both HDD and SSD. The design used on line dynamic data from track messages at the host and database objects (Tables, views, triggers, procedures) that executed at RDBMS kernel, providing robust performance and functionality.

Features of real time DBMS is used for fast DBMS transaction within deterministic criterion are still unexplored in mission critical network centric application. Development have started in Air Defense system using Oracle database / other open system databases having conventional DBMS software for supporting host data processing modules to enable better improvement in data integrity compared to flat, unrelated and unmanaged data structures. Presently US Navy’s multi input tracking and control system1 runs in a multitasking operating system but not a real-time operating system by the deterministic criterion. Hence optimizing conventional DBMS with SSD will comply better design and improve dynamic track kinematics within deterministic criterion.

During creation of Flight vehicles (FVs) and decoys, debris from PBTs, the threat initially consists of closely spaced objects. Hence it is required to identify the actual FV in the closely space scenario. The association process can only estimate the state vector of tracks and maintain it but cannot identify FV and decoy tracks. There is high volume of data access and computational requirement of kinematic parameters while dealing with the tracks processing for identification of these tracks of clusters which travels in parallel along with FV track. Figure 1 shows the incoming raw data is decoded to Engineering values in ECEF and instantly inserted into Table T1. Simultaneous it is converted into East North Vertical (ENV) value in table T2. It is further processed using tables T3 and T4. Final value of FV, decoy is obtained at every cluster group processing instantly during post boost period. Figure 2(I) and (II) give the results of improved processing time at database using SSD of two R&D trials. The experiment was done in Linux using RHEL 4.0 and Oracle 10.2 server. In high end system the results with SSD can be much improved.



Processing raw data

to engg.value at host

Data insertion to DB T1

----------- (ECEF) --------

Data insertion to DB T2

----------- (ENV) -------------

View of DB T 2 (All latest updates) **T**now, **T**now-1…

Data insertion to DB T 3

Rv, Decoy info. To T 4

(Max. and next to max. Energy)

Procedure (New Rep.)

Procedure

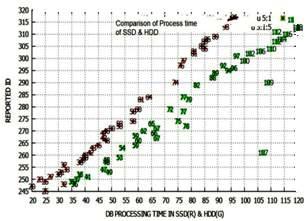
T1 ends

T1 starts

Trigger

**Fig.2 (I) above shows Comparison of HDD &**

**SSD processing time of Trial 2 (Red SSD)**



**Figure 1**

**Database design for**

**FV and decoy**

**FFig.2 (II) above shows Comparison of HDD and SSD processing time of Trial 2 (Red SSD)**

New measurements are to be compared with the old tracks for dynamic evaluation of flight path angle between new measurement and tracks, least distance of tracks to the flight plane and distance between tracks within each kinematic check. This has been analyzed in section IV. The post booster cluster groups are formed into cluster segments / groups of entire track propagation of the trajectory profile which are the movement of tracks within cluster groups. Once a track vector during kinematic checks does not commensurate within limit value, it fails to pass and only those passed track ids within last cluster group and its updates are the basis of FV and decoy

determination forming parallel trajectories. This identification is considered at every occasion of formation of last post boost cluster group till next cluster group forms when new track is reported. Presently FV and decoy is determined based on determination of highest average energy tracks and next below within each cluster group of trajectory. It is seen that the processing time in Solid State Drive rather than disk access has greatly reduced. This paper describes the benefit to use a DBMS system and shows results to reflect how optimized non real time data base can even better support the command control application with fast processing.

1. LITERATURE REVIEW

Some studies on application of database architecture and experimentations for mission critical applications, tactical defence application around the Globe have been carried out.

Ceruti and Gessay(1998)**2**, publishedpaper which has given a broader outline and comparison in order to evaluate the state of new data-access technology, methods and architectures, to explore the systems and software developed within the C4I sector that can use this technology and to recommend an architecture on which to base the framework of data-access methodologies into which Naval C4I systems could evolve.

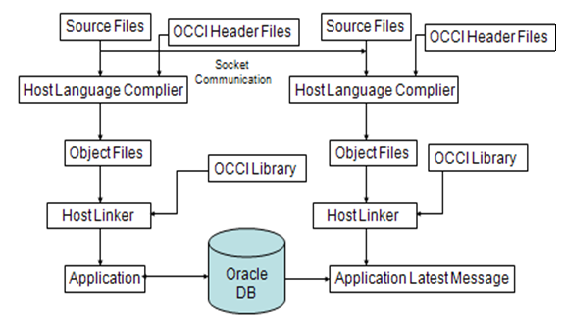
Layman and Daly (2002)**3** described in their Public report organized by Naval Research Laboratory, US Washington, DC. This paper reports the progress on the US Navy’s Embedded Simulation Infrastructure (ESI) Program implementation and management of embedded models and simulations for C4I. The Defense Information Infrastructure Common Operating Environment (DIICOE) is the framework upon which many modern U.S C4I systems are developed.

Ceruti (2003)**5**, reported in his paper explores challenges facing information system professionals in the management of data and knowledge in the Department of Defense (DOD),USA, particularly in the information systems utilized to support C4I. Para 8.2 of this report specifies architecture of dynamic database of data-driven collection management and situational estimation.

McDaniel and Schaefer (2003)**1** published report used algorithms, data characteristics, and scenarios from deployed and R&D systems. The results do show that even under intense scenarios and with massive fusion on a general purpose medium performance off-the shelf computer using a non-real time operating system, the embedded DBMS can perform adequately.

Shane and Hollenbeck (2004)**4** described in their paper a working prototype system which can help solve the problem for US department of Defence and potentially for multinational forces by incorporating a national repository and intelligent automation system using database in ACSIS system. The ACSIS is using Oracle database because Oracle is the only DBMS which incorporates EU Common Criteria certified multi-level security features, reinforcing the team’s original choice of a DBMS in designing the data model.

Above studies explores views on data access architecture on various mission critical applications on C4I and network centric areas to enable the DOD to meet the present and future information management challenges. It is to infer that while most apply in organizational contexts, the relationships derived from the view by McDaniel D. and Schaefer G. (2003)**1** is probably more appropriate for a specific application of mission critical command and control. The present paper submitted is a specific mission critical application on a non real time operating system and database where the methodology adopted, the main analysis technique employed are the main findings of the study.

****

1. METHODOLOGY

In client server of Oracle Data base model, transactions are accompanied between server and client DB where separate client installation is done at client host. In the present experiment instead of separate client installation, transaction was made by Oracle instant client where operating (Linux or other OS) can communicate with the server through OCCI library and host linker.

Fig 3 shows how data base instant client can be connected to multiple hosts having OS specific OCCI libraries. Thus the query, transactions to the data base server from host OS were accompanied by embedded ‘SQL’, ’procedure’ calls.

This is observed that process time in database shown in table 1 in the result section V based on the flow design shown in fig. 1 called through SQL call, procedure, etc. The total time for all those new 43 track processing and

further update track maintenance were counted for 21657 track occurrences in the database with a total processing time was 53082700 micro seconds thus average transaction time / row processing is 2452 microseconds in the Linux system with SSD Drive for a particular R&D trial.

**Fig. 3**

**Architecture of hosts to share DB instant**

The SQL call from host terminal created Oracle processes to execute as operating system threads. In threaded mode, some background processes on UNIX and Linux run as processes containing one thread, whereas the remaining Oracle processes run as threads within processes. Design and writing code for multi processing through multithreading in present C++ code is combursum as it required programming to create code for every sub processes / threads separately including semaphore. Whereas sharing of multi processes for multi tasking and creating threads in Oracle database is more convenient and system built in. One / more processes for each connected user or one or more processes of other user from hosts can be handled and shared by client requests with automatic shared server process in database server. Augmentation of multi processes / multi threading automated features in Oracle database has made it more convenient for real time processing. A small number of shared server processes can perform the same amount of processing as many dedicated server processes, and the amount of memory required for each user is relatively small. In a shared server connection, many client processes access a single server process having single / multiple sessions.

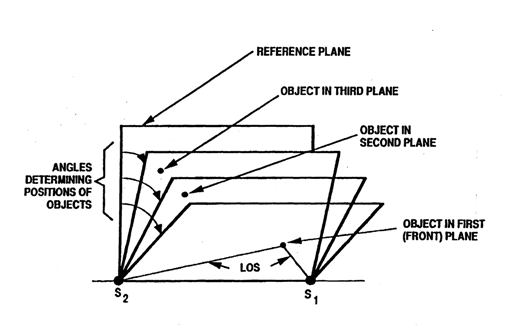
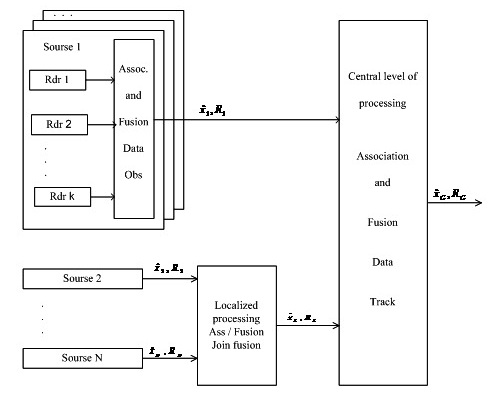
Thus application-dependent data structures of present design in Command Control subsystem algorithm application were converted into conventional DBMS’s with near real time features. Further benefits added are as such;

1. a. Data consistency, back history and auditing. Air situation can be better handled due to retrieval of back history from data base.

b. Structural data integrity. Non-redundancy which support relational normalization technique to avoid redundant storage of values.

c. Referential integrity rules, Relational DBMS implement rules that prevent fragmented objects. This will cause a complete cleanup of the data once the key component is deleted. This prevents ‘ghost tracks’ and software crashes caused the program tries to access components of an incompletely deleted or created track that no longer exists**1**.

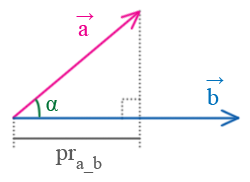
1. DATA ANALYSIS

**Figure 5**B,

**Figure 4,** Target tracking Architecture

**Kinematic checks for cluster Identification**

****

Projection from reference plane is determined to derive the cluster kinematic conditions within the cluster group during new report (Ref. Fig 5A and 5B).

The cluster track kinematic parameter checks determines FV and decoy which are processed in Oracle database. Let’s consider non collinear vectors =(xa,ya,za) and =(xb,yb,zb); let α be the angle between them (fig. 6b above). The pra\_b of

the vector on the vector is calculated in the following way:

**Figure 5**A **Proj. of vector A on vector B**



**Figure 6**

**Flowchart of cluster group & FV determination**

Flowchart of cluster group & FV determination.

Yes

No

Old Trk ? Yes /No

*R*

*E*

*T*

*U*

*R*

*N*

*N*

*E*

*X*

*T*

*N*

*E*

*W*

C

l

u

s

t

e

r

Stop

New Track Report (raw) (ECEF)

Convert to ENV ECEF

1st clust ggrp. Grp.

Last clust. ggGGGrp.(n) + 1

Check Kinematic conditions

Failed trk Failed clusters

Passed Trk Passed clusters

Alt, Av.energy, Highest (FV), next Highest En. En.(decoy)

Dynamic storage, Update at T0, T\_1 …..

This formula is obtained from correlation



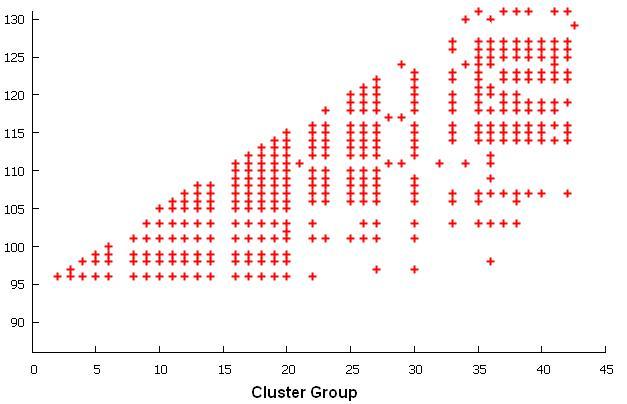
**Track Ids**



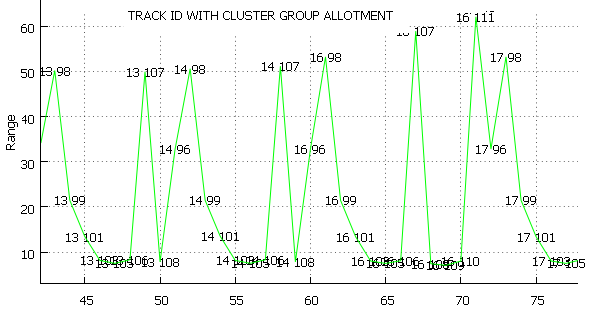
4*A*. *Track Ids in Cluster Group and Update*

Sensor level cluster group checks are done during reception of information. The track clusters are

associated from same type of radar sources and also from different local sensor sources. Associated and fused data from these two types of sensor sources are again processed by central level of processing. Sometimes fusion of data, received from numerous radar sources, based on selection of priority (main) source, performs integration of such parameters like classification of cluster tracks, state identity on target because these data are considered to be the most

 **Figure-7 Growth of Tracks in Cluster Groups. Linear growth of tracks in the group as more cluster tracks would only grow linearly with the number of targets.**

accurate in terms of tracking and conducting of combat operations. This approach does not require a number of calculations for performance of fusion and provides tracking of fused track which is good enough.



**T**

**r**

**a**

**c**

**k**

**Ids**

**Time (Sec)**

**Figure 8, A Range (Not to scale) Time plot depicts cluster group and tracks propagation along with range and time.**

Especially good track tracking is provided, if the priority source performs a complete cycle of trajectory data processing, including data fusion and smoothing. In Fig 8 above, it may be seen from the **plot** **lines** that upon receiving new track, previous tracks which were passed by kinematic checks were processed serially in one cluster group and continue to update till next track reports. On further propagation while new track is reported new cluster group forms with new set of track ids and updated. Hence highest energy FV can be determined amongst different trajectories at various ranges of time in space. Thus with new track id reported the cluster group no. has been incremented with a set of tracks included as determined by the algorithm which were updated further till next allotment.

Present design FV and decoy has been determined within a cluster group which is completely defined by specifying the set of parameters within group.

Present experiments on trial data was smoothed fused trajectory data, received from radars of the same type and used the method of integrated processing.

The integrated processing of trajectory data includes estimation of target position and motion parameters on the basis of all received data, weighted with account for the covariance matrix of errors in measurement of target coordinates and motion parameters. Hence the present cluster design in database has different concept than the sensor level cluster group. Present cluster grouping is done in database from the fusion output of a typical command control system where cluster state vector on post boost phase separation is determined and maintained by association. If the entire set of targets and measurements can be divided into sets of independent cluster groups in post boost phase, then a great deal of simplification may result. Instead of one large tracking problem, a number of smaller tracking problems and FV, Decoy determination can be solved independently within individual group. Since the amount of computer storage and computation time grows exponentially with the number of targets, this can have an important effect in reducing computer requirements.



*4B. Track Energy determination*

Algorithm used in simulation study for calculation of highest energy calculation **7** (P 18) has been used for calculation of track energy at the latest update of track comparison for FV and number of decoys in Oracle database and the two samples amongst last cluster group has been evaluated for FV and decoy. The total energy of a ballistic object is unchanged throughout its exo atmospheric flight. ‘HE’ denote flight vehicle is

XL

*Local*

*Meridian*

ZE

XE

P1(x1 ,y1 ,z1)

**(0;0;0)**

N-pole

ZL

YL

YE



**Figure 9 ECEF and ENV frame**

point P1. Alt = [p/cos ()] – R, -with reference to sealevel.

**S**emi major axis , Semi minor axis 

=0.00669437999, =0.00673949674222

e1 and e2 – eccentricities of the Earth ellipsoid.

P1 is the position vector  .  -Target

velocity vector in a earth centered Cartesian coordinate system.

-is the latitude at point P1.

Hence HE energy equation can be derived as

equal to sum of the energy (**Ek**) and potential energy **(Ep).** VT is the target velocity, **g** the gravitational constant, R the radius of the earth and RT the distance between the target and the center of the earth. Radius of the =a / (1-e12sin2)1/2 - effective radius of earth at

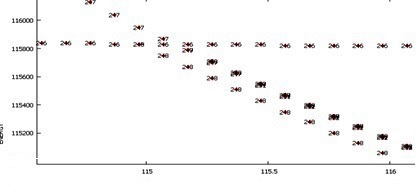


1. RESULTS

TABLE 1

DATABASE PROCESS TIME OF SIMULATION

|  |  |  |  |
| --- | --- | --- | --- |
| **Id** | **Id\_cnt** | **Proc\_t Tr/row(mics)** | |
| 245 | 361 | 80275 | 222 |
| 247 | 188 | 66600 | 354 |
| 248 | 191 | 86425 | 452 |
| 249 | 57 | 23300 | 409 |
| 250 | 169 | 75800 | 449 |
| 251 | 35 | 14950 | 427 |
| 252 | 358 | 184700 | 516 |
| 253 | 57 | 26600 | 467 |
| ---- | ----- | ------ | ------ |
| 305 | 7 | 7175 | 1025 |
| 307 | 139 | 130850 | 941 |
| 308 | 59 | 54100 | 917 |
| 309 | 344 | 361700 | 1051 |
| 313 | 145 | 145450 | 1003 |
| 315 | 176 | 184300 | 1047 |
| 316 | 204 | 209375 | 1026 |
| 317 | 15240 | 48160925 | 3160 |
| T=43 | 21657 | 53082700 | 2452 |



**Energy**

**Figure 10 Time and Energy of fragments of a simulated trajectory portion. In figure 10, the FV and decoy booster has been determined at various time instant.**

From various R&D trials and from simulation, the result of one R&D trial is tabulated in Table 1 for a flight vehicle and its decoy/booster has been shown in microseconds for receiving a total of 43 tracks in

**Time**

Oracle database. However due to error associated with the track parameters FV in the trajectory can be

distinguished if FV energy>=1.3 decoy energy. The period of a cluster group is the time for forming a group during new track measurement and its update period. Hence it will be helpful for BMD to obtain the FV information for engagement at the required moment of the mid course trajectory.

TABLE 2 FLIGHT VEHICLES AND DECOY\*

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **FLIGHT VEHICLE** | | | **DECOY** | | **FLIGHT VEHICLE** | | | | **DECOY** | |
| **TIME** | **He MAX** | **ID** | **He MAX** | **ID** | **TIME** | | **He MAX** | **ID** | **He MAX** | **ID** |
| **113.87** | 115850 | 245 | ------ | ------- | **116.57** | | 115810 | 246 | 114780 | 249 |
| **114.37** | 116500 | 247 | 115840 | 246 | **116.77** | | 115810 | 246 | 114660 | 249 |
| **114.07** | 114.07 | 247 | 115850 | 245 | **116.97** | | 115810 | 246 | 114540 | 249 |
| **114.97** | 115950 | 247 | 115830 | 246 | **117.17** | | 115810 | 246 | 114420 | 247 |
| **115.17** | 115830 | 246 | 115790 | 247 | **117.27** | | 115810 | 246 | 114370 | 249 |
| **115.27** | 115830 | 246 | 115830 | 246 | **117.47** | | 115810 | 246 | 114260 | 248 |
| **115.47** | 115550 | 247 | 115820 | 246 | **117.57** | | 115810 | 246 | 114210 | 249 |
| **115.67** | 115400 | 249 | 115820 | 246 | **118.57** | 115810 | | 246 | 113740 | 249 |
| **116.07** | 115110 | 249 | 115820 | 246 | **118.77** | 115810 | | 246 | 113650 | 249 |
| **116.27** | 114970 | 247 | 114970 | 247 | **118.97** | 115810 | | 246 | 113570 | 249 |
| **116.37** | *1*15820 | 246 | 114910 | 249 | **119.27** | 115810 | | 246 | 113450 | 249 |

Table 2 shows the relation of highest energy (FV) and (decoy) next below highest energy tracks determined as per above equation (2) within cluster groups till next cluster group id forms. This will be convenient to find out the FV at every segment of trajectory of track propagation

VI.CONCLUSION AND FUTURE SCOPE

The commercial Database (Oracle DB) has achieved the near real-time database (RTDB) property whose operations execute with predictable response, and with application-acceptable levels in addition to timely execution of transactions. Powerful information access by advanced SQL pass-through method to database and techniques with embedded C++ program & tools can provide a foundation for better air situation processing. Because the design is transparent, many types of air situation algorithm can be unified within it and operate in an integrated manner over a broad spectrum of information processing. The data base resources and objects can be shared that they will interoperate with other integrated system in a collaborative manner. Present platform centric war fighting can be realized and transformed to Network centric war fighting based on information fusion and the methodology presently used which is based on dynamic data access against track situation awareness. As the complexity of real-time systems and application is going up, the amount of information to be handled by real-time systems increases, motivating the need for database and data service functionality (as opposed to ad hoc techniques

and internal data structures), the results of the experiments provide evidence and argues that; apart from high-performance embedded DBMS’s which have much higher data access and quick analysis capability than application dependent data structures, a non real time database can also be optimized and designed which can perform under intense scenarios the air situation process on a general purpose medium performance computer using a non-real-time operating system, the embedded DBMS can perform adequately better from those RDBMS vendors which has advanced SQL capabilities with proprietary implementation of stored objects (procedures, functions, etc.). This advanced SQL is comprised of variables, procedures, functions, etc. that execute at the RDBMS kernel, providing robust performance and functionality. This type of RDBMS databases (e.g. Oracle) can be optimized with Solid State Drive with non real time based Operating system to go in a long way. With more powerful processors, a real-time operating system, and a distributed computing environment, the embedded DBMS could be expected to perform even better.

ACKNOWLEDGEMENT

The author would like to express his sincere appreciation for embedded programming support by collogues of DCCT/C4I, of DRDL. This application would not have completed without their support, discussion. This division has not only intelligent programmer, but domain expert in the field of BMD. The knowledge of that has contributed much to the successful development of cluster processing algorithms.

REFERENCES

[1] McDaniel D. and Schaefer G. (2003), Real-time DBMS for Data Fusion, proceedings of the sixth international conference ,IEEE conference publication, Vol. 2, pp 1334-1341.

[2] Ceruti M. G and Gessay S. A (1998), White paper on Next-Generation Data-Access Architecture for Naval V4I Systems, pp 1-24.

[3] Layman G. and Daly J. (2002), C4I Tactical Applications Utilizing Embedded Simulations, pp 1-15, organized by Naval Research Laboratory, Washington, DC

[4] Shane R. and Hallenbeck P. (2004), Proceedings of 9th International Command and Control Research and Technology.

[5] Ceruti Marion G (2003), (pp1-12), Report on “Data Management Challenges and Development for Military Information Systems, DOD , Govt. of US”, Published in *IEEE Transactions on Knowledge and Data Engineering (TKDE).* Vol. 15, No. 5, pp. 1059-1068, September/October 2003.

[6] Oracle documentation for multi process and multi threading,

[7] Chang C B, (1979-1), Technical note Optimal State Estimation of Ballistic trajectories with Angle-Only Measurements,

Ballistic Missile Defense Program Office, Department of the Army,.

MA **22133**