# *CADAC REAL-TIME DOCUMENTATION*

VOLUME IV

**Release 3.0**
**September 1998**

**Prepared For:**
*Air Force Research Laboratory (AFRL/MNG)*
*Eglin Air Force Base, Florida 32542*

**Prepared By:**
*TYBRIN Corporation*
*745D Beal Parkway NW*
*Ft. Walton Beach, FL 32547*

**DISTRIBUTION UNLIMITED**

## Table of Contents

## List of Figures

## List of Tables

# 1. INTRODUCTION

Program CADAC, Computer Aided Design of Aerospace Concepts, provides an environment for the development of general purpose, digital computer simulations of time phased dynamic systems. It manages input and output, generates stochastic noise sources, controls all state variable integration and provides post-processing data analysis and display. CADAC has proven its adaptability to many simulation tasks: air-to-ground weapons, air-to-air missiles, ground-to-space and space-to-ground vehicles, and airplanes. The CADAC environment is suitable for 3DOF, 5DOF, and 6DOF simulations. It supports deterministic and Monte Carlo runs. Output can be listed or plotted.

This version (3.0) is a revision of the second release of CADAC for Personal Computers (August 1998). The distribution of CADAC is unrestricted, however, the US Government does not warrant the product nor does it accept any liability for its use.

The documentation consists of four parts, ***Quick Start***, ***User Documentation***, ***Program Documentation*** and ***Real-time CADAC Documentation***. They are written in Microsoft Word 97 and can be found on the ftp site under the filenames QUICK30.DOC, USER30.DOC, CADAC30.DOC and RTCAD30.DOC. For the novice who wants to get a quick overview of CADAC and run the supplied test case ***Quick Start*** is the only document needed to read**.** ***Quick Start*** provides the experienced user with file lists and schematics that serve as quick reference for simulation modifications.

The ***User Documentation*** addresses all capabilities of the CADAC development environment. It should give answers to most questions that come up during the design of a new trajectory simulation. Summary tables examples and matrix utilities provide useful references. The serious CADAC user should read this entire document

The ***Program Documentation*** provides greater detail for many subjects from building the input and the header files, the integration routine, the generation of stochastic variables, the execution of multi-runs, sweep runs, or Monte Carlo runs to the utilities that aid in building, documenting and analyzing CADAC simulations. It should be used as a reference as more specific questions arise.

The ***Real-time CADAC Documentation*** (this report) addresses all the functionality and capabilities involved in the real-time CADAC methodology. It provides detailed procedures in executing CADAC to generate the necessary data files needed to produce a real-time version of the modules. Step-by-step instructions are also detailed for executing the CONVRT program, which create the real-time CADAC code used in a simulator or a test bed.

## 1.1 Real-time CADAC Objective

The objective for implementing the real-time CADAC methodology extends to simplifying the introduction of a new missile concept into air combat simulators. The ultimate objective is to provide a subroutine that is included in the simulator without any modifications to the missile subroutines. The interaction between the real-time CADAC routine and the simulator should also be as simple as possible. By simplifying the introduction of missile subroutines into the simulator, a point-mass type model can be replaced with a full 6 DOF simulation to model high angle of attack effects caused by either the super-maneuverable aircraft launches or extreme target maneuvers. The implementation of the real-time CADAC methodology facilitates the inclusion of CADAC models into air combat simulators.

### 1.2 Real-time CADAC Overview

CADAC is a missile trajectory simulation written in FORTRAN consisting of 5 and 6 DOF models. CADAC also provides modeling target maneuvers, although the maneuvers are defined prior to missile launch.

The 5 DOF version is commonly referred to as the psuedo 5 DOF implementation modeling 3 translational and 2 rotational degrees of freedom. Three translational degrees of freedom are modeled by nonlinear differential equations employing tabulated trimmed aerodynamic data. The two rotational degrees are pitch and yaw (skid-to-turn) or pitch and roll (bank-to-turn). They are modeled by linearized differential equations that describe the attitude dynamics of the missile. In this case, Euler's equations are not solved. The 6 DOF version is a full 6 DOF simulation solving the three translational degrees with Newton's equations and the three rotational degrees with Euler's equations.

The real-time CADAC application consists of a single missile subroutine with the target and data link information removed. This subroutine is a self-contained file that can be incorporated into a MIL simulator or can be executed as a check-out test bed prior to inclusion in the MIL simulator. The target and data link information is provided to the real-time subroutine by the MIL frame or by the TRACK.ASC file created using batch CADAC. The target track data may be provided at a slower time increment and therefore needs to be extrapolated for the current CADAC time; this is done at each CADAC integration time. The extrapolation is performed by assuming a second order polynomial fit for the target track data and then using the polynomial coefficients to extrapolate the data for the current CADAC time.

### 1.3 CADAC Architecture

CADAC simulations consist of modules that represent the major subsystems of an aerospace vehicle. The interface between modules are controlled by equivalencing the variables to a global common array C(3510). The module structure for 5 and 6 DOF air-to-air missile simulations are shown in Figure 1 and 2, respectively.

The first two characters in the blocks identify the module, followed by the first C array location reserved for that module (see Table 1). For all modules, 100 adjacent locations are reserved with the exception of the D2 (Euler's equations) module, reserved locations 1700–1749, and the G4 (end-of-run) module, reserved locations 1750-1799. Mode switches (shown beneath the module blocks in Figures 1 and 2) control some of the modules. The calling sequence is important and must be maintained for the simulation. For real-time applications, the two darkened blocks in Figures 1 and 2 (modules G1 Target and S2 Radar) are removed and the input provided by these modules are replaced by the MIL frame. All 5 and 6 DOF missile models are composed of the same set of module names and interfaces.

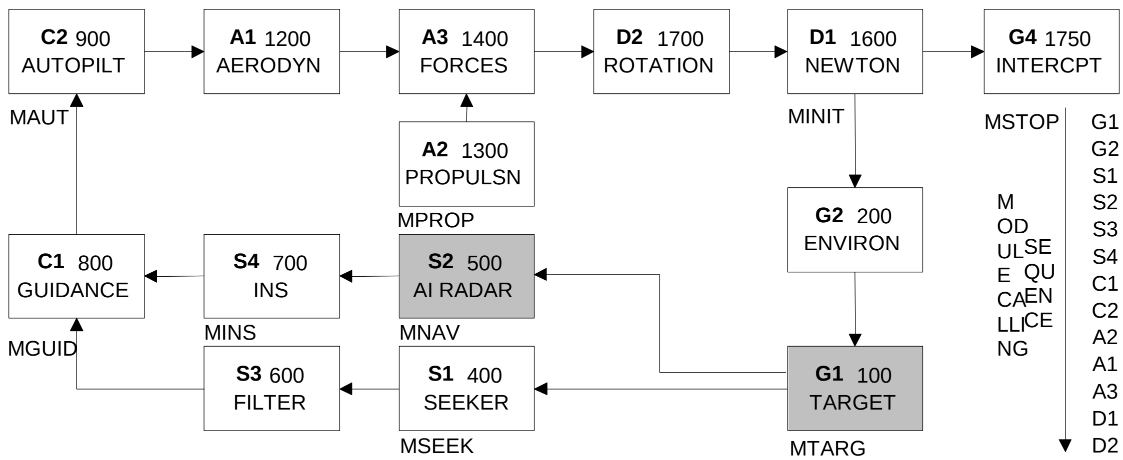


**Figure 1. 5 DOF CADAC Missile Simulation.**

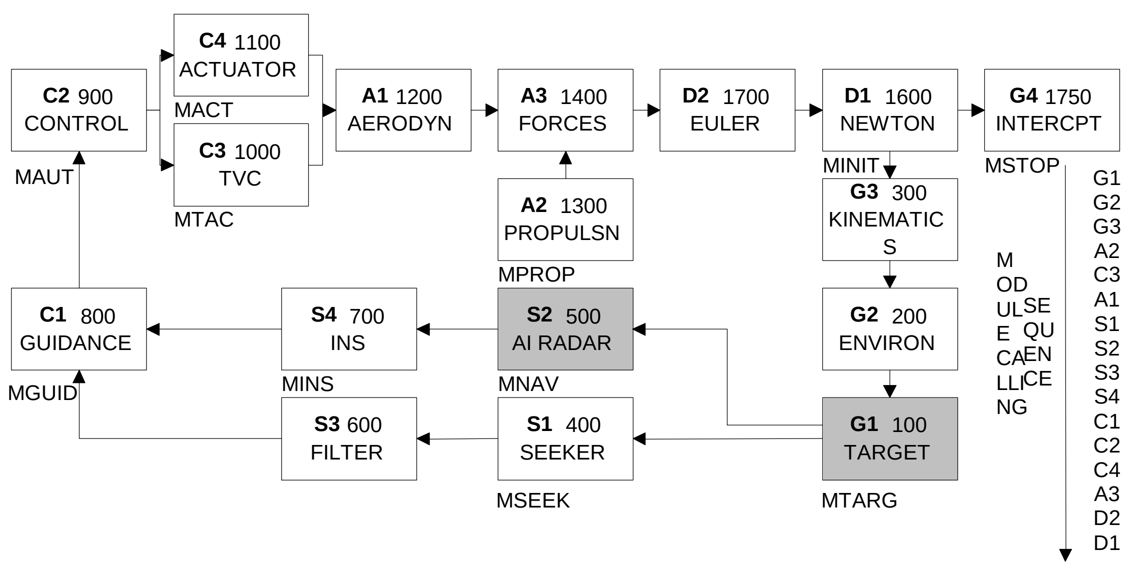


**Figure 2. 6 DOF CADAC Missile Simulation.**

**Table 1. Module Names and Reserved Common Locations.**

| MODULE | MODULE NAME | RESERVED COMMON LOCATIONS | MODULE NUMBER |
|---|---|---|---|
| A1 | Aero Coefficients | 1200 - 1299 | 2 |
| A2 | Propulsion | 1300 - 1399 | 3 |
| A3 | Forces & Moments | 1400 - 1499 | 4 |
| A4 | | 1500 - 1599 | 5 |
| C1 | Guidance | 800 - 899 | 7 |
| C2 | Auto pilot | 900 - 999 | 8 |
| C3 | TVC | 1000 - 1099 | 9 |
| C4 | Actuator | 1100 - 1199 | 10 |
| D1 | Newton Eqs. | 1600 - 1699 | 17 |
| D2 | Euler Eqs. | 1700 - 1799 | 18 |
| D3 | | 1800 - 1899 | 19 |
| D4 | | 1900 - 1999 | 20 |
| G1 | Target | 100-199 | 22 |
| G2 | Air Data | 200-299 | 23 |
| G3 | Kinematics | 300-399 | 24 |
| G4 | Terminal | 1750-1799 | 25 |
| G5 | | | 26 |
| S1 | Seeker Model | 400-499 | 28 |
| S2 | Sensor Model | 500-599 | 29 |
| S3 | NAV Filter | 600-699 | 30 |
| S4 | INS Model | 700-799 | 31 |

# 2. BATCH CADAC CONVERSION

In order to create a real-time application of a given missile simulation, the batch CADAC version should be executed first to create the necessary target data files. The batch CADAC program must be prepared to generate two real-time input files: INIT.ASC and TRACK. The HEAD.ASC file must be marked to identify the variables to be written to the initialization file INIT.ASC and the target track and data link file TRACK.ASC. With these items set in the HEAD.ASC file, executing the batch CADAC model will create the necessary real-time input files. The converter program using the batch CADAC files as input can create the real-time CADAC file: MODULES.FOR, CADIN.ASC, HEAD.ASC and INIT.ASC. The result of executing the converter program is a FORTRAN coded file that when linked with the CADAC matrix utilities file UTL.FOR can be used for inclusion in the MIL frame or used for check-out in a test bed. This document describes in more detail this procedure for creating a real-time version of the missile simulation modeled by the batch CADAC version.

## 2.1 Batch CADAC

The batch CADAC model is the simulation created using the instructions described in the ***CADAC User Documentation***. The batch CADAC model requires two input files, CADIN.ASC and HEAD.ASC, that has been properly prepared to create real-time files, INIT.ASC and TRACK.ASC. The converter program and the real-time test bed will use these files along with the modules.

### 2.1.1 CADIN.ASC

The CADIN file is a fixed-formatted file that directs the CADAC missile trajectory. This file contains information regarding the module calling sequence, data initialization, trajectory staging and simulation integration time step. This file also controls the print rates for CADAC generated output files. The TRACK.ASC file is a time dependent data file containing the target and data link update information. The print rate to this file is set in the CADIN file using the CPP (C(2015)) variable, the print rate at which the TABOUT data is written. The print rate CPP should be set to the approximate MIL frame integration time step. The use of ‘save state cards’ or ‘groups cards’ is not permitted.

### 2.1.2 HEAD.ASC

The HEAD file is the CADAC variable definition list. The file contains a list of scroll variables and plot variables. The first line (the option list) contains keywords to indicate the type of CADAC output files that are to be created. For the real-time application, the HEAD option list must include the keywords INITASC (or INITBIN for binary file) and TRACKASC (or TRACKBIN for binary file). The INITASC keyword instructs CADAC to create an initialization file INIT.ASC while TRACKASC indicates the creation of the target track and data link file TRACK.ASC.

The HEAD file also indicates which variables are to be output to CADAC files. For the real-time application, two types of variables are needed: Initialization and Tracking. Placing an ‘I’ in column 1 of the HEAD plot variable list marks the initialization variables. Placing a ‘T’ in column 1 marks tracked variables. If the variable needs to be both an initialization and tracked variable, then a ‘B’ is placed in column 1. If the variable is an array the array must be expanded and the individual elements of the array are marked. As always, integer variables carry an ‘I’ in column 3. By following these rules, the necessary output files for real-time CADAC are created.

The variables marked for output to the real-time files are considered 5 DOF or 6 DOF variables are depicted in Table 2. ‘I’ indicates the variables that must be initialized due to the removal of the G1 modules. Variables that are provided in a continuous stream are called track variables as indicated by a ‘T’. They replace input that would have come in part from the removed G1 and S2 modules. Some are state variables and therefore must also be initialized and are marked by a ‘B’.

### 2.1.3 INIT.ASC

The INIT data file contains data in a format similar to the other CADAC plot files TRAJ and STAT. Data for up to 70 variables may be stored on the file. The variable data printed on this file is selected through the HEAD.ASC input file. The first line of the file is a title record. This record is a 100-character string. The format of an ASCII file requires that the first character in the title be a “1”. The binary file may have any other character as the first character, but a “ “ is preferred. Following the title record is a record of three integers. The integers written to the file are C locations 1982, 1983 and the executive variable NPLOTVAR. Variable NPLOTVAR indicates the number of variables whose data is written to the file.

Following the integer data are the character acronyms for the variables whose data is written to the file. These acronyms are 8 characters in length and are separated by a string of 8 blanks. The variable data follows the acronym data. The data is printed in this file only once, after the initialization modules have been processed and before the integration begins. Time is always the first variable printed to INIT by CADAC.

The data on this file represents the initialization of the target state variables and data link, output generated by subroutines G1I and S2I, respectively. This data file is required input for the converter program, but is not necessary for the real-time CADAC test bed.

### 2.1.4 TRACK.ASC

The TRACK data file contains data in a format similar to the other CADAC plot files TRAJ and STAT. Data for up to 70 variables may be stored on the file. The variable data printed on this file is selected through the HEAD.ASC input file. The print rate of the data to the file is taken from the global variable CPP (C location 2015). The first line of the file is a title record. This record is a 100-character string. The format of an ASCII file requires that the first character in the title be a “1”. The binary file may have any other character as the first character, but a “ “ is preferred. Following the title record is a record of three integers. The integers written to the file are C locations 1982, 1983 and the executive variable NPLOTVAR. Variable NPLOTVAR indicates the number of variables whose data is written to the file.

**Table 2. Initialization Variables for 5 and 6 DOF Models.**

| 5 DOF Variables | | |
|---|---|---|
| Variable Name | B Array Location | Definition |
| ALP | 0943 | I Angle-of-attack – rad |
| BET | 0947 | I Sideslip angle – rad |
| SBEL(3) | 1602-1604 | I Missile position wrt point E in L coord – m |
| VBEL(3) | 1605-1607 | I Velocity of missile in L coord – m/s |
| HBE | 1615 | I Missile height above sea level – m |
| DVBE | 1641 | I Missile speed – m/s |
| PSIVL | 1643 | I Horizontal flight path angle – rad |
| THTVL | 1645 | I Vertical flight path angle – rad |
| SBELS(3) | 1649-1651 | I State variable of missile position in L coord – m |
| VT1EL(3) | 0119-0121 | B Velocity of target wrt earth in L coord – m/s |
| ST1EL(3) | 0125-0127 | B Position of target wrt earth reference point E in L coord – m |
| VT2EL(3) | 0169-0171 | B Velocity of shooter wrt earth reference point E in L coord – m/s |
| ST2EL(3) | 0175-0177 | B Position of shooter wrt earth reference point E in L coord – m |
| TSIGN | 0402 | B Target signature variable |
| TT1L(3,3) | 0130-0138 | T Tranformation matrix of target body coord wrt L coord |
| MNAV | 0500 | T Timing flag; =0: Reset, =1: Track, =2: Measured, =3: Update |
| ST1CEL(3) | 0511-0513 | T Target position wrt E measured by AI radar in L coord – m |
| VT1CEL(3) | 0514-0516 | T Target velocity measured by AI radar in L coord – m/s |
| 6 DOF Variables | | |
| Variable Name | B Array Location | Definition |
| PSIBLX | 0337 | I Yawing angle of missile – deg |
| THTBLX | 0338 | I Pitching angle of missile – deg |
| PHIBLX | 0339 | I Rolling Angle of missile – deg |
| VBEB(3) | 1613-1615 | I Missile velocity in body coord – m/s |
| SBEL(3) | 1619-1621 | I Missile position wrt point E in L coord – m |
| DVBE | 1636 | I Missile speed – m/s |
| HBE | 1639 | I Missile height above sea level – m |
| PP | 1705 | I Body roll angular velocity – rad/s |
| QQ | 1707 | I Body pitch angular velocity – rad/s |
| RR | 1709 | I Body yaw angular velocity – rad/s |
| VT1EL(3) | 0119-0121 | B Velocity of target wrt earth in L coord – m/s |
| ST1EL(3) | 0125-0127 | B Position of target wrt earth reference point E in L coord – m |
| VT2EL(3) | 0169-0171 | B Velocity of shooter wrt earth reference point E in L coord – m/s |
| ST2EL(3) | 0175-0177 | B Position of shooter wrt earth reference point E in L coord – m |
| TSIGN | 0402 | B Target signature variable |
| TT1L(3,3) | 0130-0138 | T Tranformation matrix of target body coord wrt L coord |
| MNAV | 0500 | T Timing flag; =0: Reset, =1: Track, =2: Measured, =3: Update |
| ST1CEL(3) | 0511-0513 | T Target position wrt E measured by AI radar in L coord – m |
| VT1CEL(3) | 0514-0516 | T Target velocity measured by AI radar in L coord – m/s |

NOTE: Vectors must be expanded and individual elements marked accordingly

Following the integer data are the character acronyms for the variables whose data is written to the file. These acronyms are 8 characters in length and are separated by a string of 8 blanks. The variable data follows the acronym data. At a given time, the value of each variable is written to the file. Time is always the first variable print to TRACK by CADAC. The data is written to the file during trajectory simulation at user-designated intervals. Unlike the TRAJ file, the TRACK file contains data for only one trajectory, since the real-time CADAC application only simulates one missile in real-time.

The converter program does not require this data file, but it is necessary for the real-time CADAC test bed. The data contained on this file replaces the information output by the target module G1 and the radar module S2; this data file simulates the update information provided by the MIL frame.

## 3. THE CONVERTER PROGRAM: CONVRT

The converter program changes the missile simulation from a CADAC batch program to a single missile subroutine suitable for real-time execution. Whether the simulation actually runs in real-time depends on the processing power of the real-time facility. However, the CADAC executive routine is stripped down to a bare minimum for use with the real-time version. An illustration of the conversion of a batch CADAC model into a real-time version is depicted in Figure 3. Sample execution of program CONVRT is described in Appendix A.

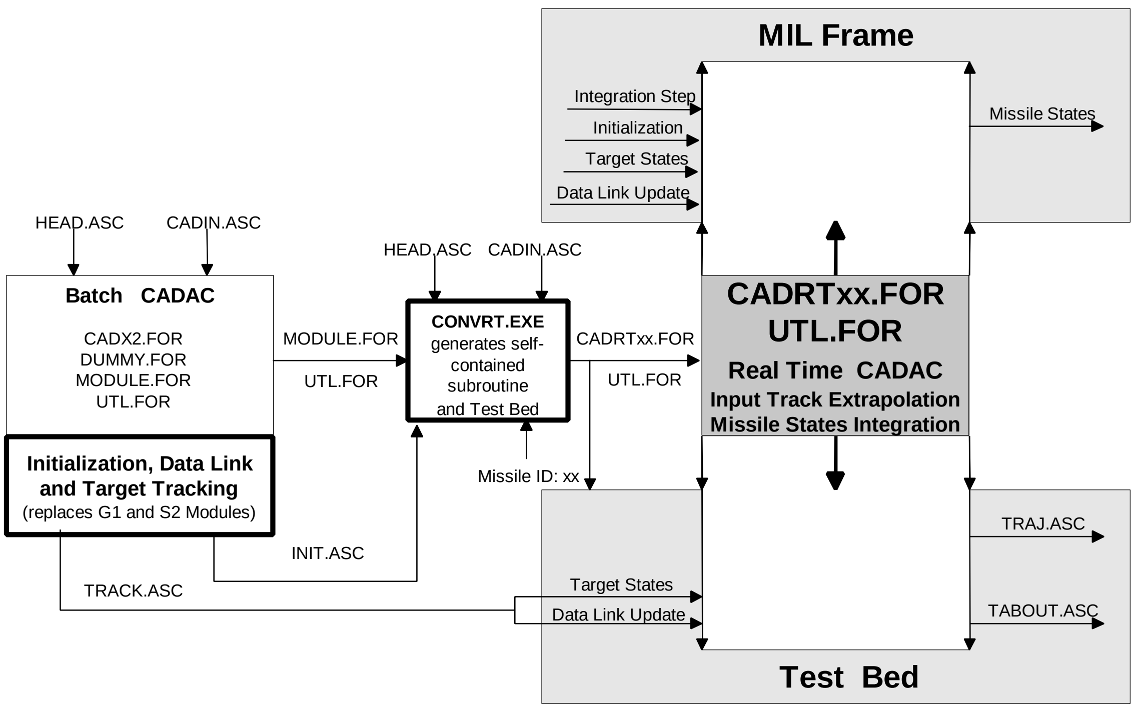


**Figure 3. Conversion of Batch CADAC to Real-time CADAC.**

The xx designation on the real-time CADAC filename identifies a particular missile. This designation is appended to all subroutine names and labeled commons within the module file. This designation is not added to the matrix utilities in the UTL.FOR file. NOTE: the real-time CADAC file must be compiled and linked with the CADAC matrix utilities file UTL.FOR for use as a real-time application.

The CONVRT program reduces the executive routine drastically for the real-time CADAC version. To take advantage of the fewer array locations, the C(3510) array is converted into a B(2620) array. (The user is cautioned against using a B array in any module subroutines.) The locations assigned in the modules remain unchanged, but the executive variables have been compacted into new locations as shown in Table 3.

**Table 3. B Array and C Array Correspondence.**

| B Array | Variable Name | C Array |
|---|---|---|
| B(1) | DTFRAME – MIL frame time step | C(2100) |
| B(2) | MSTAT – Missile status flag for MIL test | C(2101) |
| B(3) | NTEST – Number of staging tests | C(2102) |
| B(4) | TIME – Simulation time | C(2000) |
| B(5) | LCONV – Simulation end flag | C(2020) |
| B(6) | NV – Number of variables in plot list | C(2280) |
| B(7) | NJ – Number of integrated state variables minus integration frequency DER(1) | C(2285) |
| B(8) | NIP - Number of integrated state variables | C(2561) |
| B(9) | ICOOR – Integration loop flag | C(2866) |
| B(10) | KSTEP – Integration flow flag | C(2011) |
| B(11) | PCNT – Succeeding TABOUT print times | C(2003) |
| B(12) | PPNT – Succeeding plot file print times | C(2004) |
| B(13) | PPP – Plot file print rate | C(2005) |
| B(14) | CPP – TABOUT print rate | C(2015) |
| B(15) | TSTAGE – Time of previous stage | C(2001) |
| | | |
| B(2000-2099) | IPL – State variable derivative value | C(2562-2661) |
| B(2100-2199) | IPLV – State variable value | C(2867-2966) |
| B(2200-2300) | DER – DER(1) integration frequency | C(2664-2764) |
| B(2301-2401) | V – Intermediate integration values | C(2765-2865) |

## 3.1 Input Data

Executing the CONVRT program requires input from 4 files that must be present in the local directory: MODULES.FOR, CADIN.ASC, HEAD.ASC and INIT.ASC. These files provide all the information contained within the CONVRT output file CADRTxx.FOR, which can be used for inclusion into a MIL simulator or used as a check-out test bed.

The MODULES.FOR file is a user-generated code file that contains the missile simulation. The creation of the MODULES.FOR file can be generated by following the guidelines set forth in the ***CADAC User Documentation***. When this file is included in the CADRTxx.FOR file, the target and radar subroutines are removed as well as references to the routines. The xx missile designation identifier is also added to all subroutines and labeled commons. The C array is then replaced with the B array where all module locations (as described in Table1) remain unchanged but the executive array locations are changed according to Table 2. Unlike the batch CADAC version, the Dummy routines (DUMMY.FOR) are not necessary since they do not perform any function within the simulation. The 3 remaining input files are discussed in Section 2.1.1, 2.1.2 and 2.1.3, respectively.

## 3.2 Output Data: CADRTxx.FOR

Executing the CONVRT program creates a single missile subroutine CADRTxx.FOR that can be called from the MIL frame. The single missile is identified by the two digit number xx. CADRTxx.FOR is self-contained with every missile having its own standard atmosphere. It integrates missile states and extrapolates the input data from the MIL frame that may enter at a lower rate than the missile integration. Each missile starts its own clock T. The fly-out is terminated when LCONV is greater than or equal to 2.

The format of the CADRTxx.FOR file is divided into 3 sections: (1) Executive, (2) MIL Frame or Test Bed and (3) Modules. The first section is unchanged from missile simulation to missile simulation and from MIL frame version to test bed version. The data and format of the second section is dependent upon the model trajectory and initialization as described by the CADIN file and the version of the real-time CADAC code. The final section contains the user-defined modules. When the CADRTxx.FOR file is used in the MIL frame or the test bed, the CADAC matrix utilities must accompany the real-time code. For use in the test bed, these files need to be compiled and linked together prior to executing the test bed.

### 3.2.1 CADRTxx.FOR Executive Section

The Executive section of CADRTxx.FOR controls the execution of the simulation. The subroutines contained in this section are:

EXEC
LDB
LDDERV
AMRK
AUXTRK
POLCOF
POLINT

The EXEC subroutine controls the initialization of simulation global variables such as gravity (AGRAV), the radius of the earth (REARTH) and conversion constants (CRAD and CFTM among others). The integration cycles are also controlled by EXEC. Loading the state variable values and their derivatives from the integration array to the B array are called by EXEC. The integration arrays are loaded from the B array prior to the call to the integration routine AMRK while the B array is loaded from the integration array after the AMRK call. After the integration cycle is completed for one time step, EXEC calls the staging routine and determines the flow for the next integration cycle. Finally, EXEC determines if an update to the target track and data link is needed.

The LDB subroutine loads the values from the integration array into the B array. The integration arrays hold the state variable and derivative values for each integration cycle. These values must be loaded into the B array for use by the simulation modules.

The LDDERV subroutines load the integration arrays from the B array. This is done prior to the execution of the integration routine AMRK.

The AMRK subroutine is the integration routine. The integration routine is a modified Euler integration algorithm incorporating a predictor-corrector cycle. The values of the state variables and their derivatives are stored in the V and DER during these cycles.

The AUXTRK subroutine is the track data calculation routine. This routine uses the polynomial coefficients (calculated by POLCOF and POLINT) to determine the track data for the current CADAC time. The coefficients are solved using the Horner scheme:

$$y = a_0 + x(a_1 + xa_2)$$

where a0, a1 and a2 are the coefficients as provided by POLCOF and POLINT
y is the current value for the track data variable
x is the current CADAC time.

The POLCOF (polynomial coefficient calculation routine) and POLINT (rational polynomial determination routine used by POLCOF) subroutines are used to determine the polynomial coefficients for the track data information as provided by the MIL frame or TRACK.ASC. These subroutines were taken from ***Numerical Recipes*** (Press, William H., et. al. *Numerical Recipes in FORTRAN 2nd Edition*, Cambridge University Press, 1992). These routines are called during the track update. Before these routines can be used by the real-time application, a time history of values must be prepared. The history consists of the 3 most recent values for each track variable. Consequently, during the first 3 integration cycles POLCOF and POLINT are not called and no data extrapolation occurs. After the fourth integration cycle, the track and uplink data polynomial coefficients are calculated after an update has been provided. As updates are made throughout the real-time execution, the earliest of the 3 most recent values is deleted and the newest update value is added to the history list and the polynomial coefficients are recalculated. The extrapolation routine solves the polynomial based on these coefficients for the current CADAC time to determine the value for use in the modules. The extrapolation routine is called between updates at every integration cycle before the predictor loop. If the any track variable is used as an integer within the simulation, the update and extrapolation routine is not utilized, the data remains unchanged unless altered by the modules or the MIL frame/ TRACK.ASC file.

### 3.2.2 CADRTxx.FOR MIL Frame or Test Bed Section

The CADRTxx.FOR file can be used as a subroutine in a MIL simulation or as a check-out test bed. The subroutines contained within this section depend on which option is desired: MIL Frame version or test bed version. Table 4 indicates the subroutines used with each real-time CADAC version.

The CADRTxx subroutine is the first subroutine executed. This can either be the subroutine called by the MIL frame (SUBROUTINE CADRTxx) or the program start routine (PROGRAM CADRTxx) when used as a test bed. When used with the MIL frame, the module section transfers the argument data (MSTAT and MNAV) to the B array for use. The output initialization routines and the track data update routine are also called from CADRTxx. The remaining subroutines for this section are self explanatory as detailed in Table 4.

**Table 4. MIL Frame and/or Test Bed Subroutines.**

| Subroutine | Definition | MIL Frame | Test Bed |
|---|---|---|---|
| CADRTxx | Controlling real-time subroutine | ✓ | ✓ |
| AUXI | Initialization modules calling routine | ✓ | ✓ |
| AUXSUB | Main modules calling routine | ✓ | ✓ |
| EXI | Variable initialization routine | ✓ | ✓ |
| EX | Staging routine | ✓ | ✓ |
| OPNF | File open routine | | ✓ |
| OPTABI | TABOUT data initialization | | ✓ |
| AUXTI | Track data initialization routine | ✓ | ✓ |
| TRACKI | Track variable definition routine | ✓ | ✓ |
| OPTAB | TABOUT output routine | | ✓ |
| OPTRAJ | TRAJ output routine | | ✓ |
| UPDATE | Track data and uplink update routine | ✓ | ✓ |

### 3.2.3 CADRTxx.FOR Modules Section

The third section of the CADRTxx.FOR file contains the user-defined modules. Changes made to these modules include:

- Replacement of the C array with the B array
- Removal of the G1 and S2 modules
- Addition of the xx missile designation identifier to subroutine name and labeled commons

# APPENDIX A

# PROGRAM CONVRT EXECUTION

## A.1. Test Bed Check-out Version

The Batch CADAC program must first be prepared to generate the two input files INIT.ASC and TRACK.ASC (or in binary form INIT.BIN and TRACK.BIN). They are created from Batch CADAC by appending INITASC (or INITBIN) and TRACKASC (or TRACKBIN) on the option list of the HEAD.ASC file.

Input to the CONVRT program are the MODULE.FOR, HEAD.ASC, and CADIN.ASC files used during the Batch CADAC execution. Also required is target initialization file INIT.ASC.

With these files in the local directory, the CONVRT program can be executed; the program is executed by double-clicking CONVRT in the Windows Explorer or using the Execute CONVRT option from the DIGITAL Visual FORTRAN compiler after the CONVRT source has been compiled and linked.

When the program is executed, the user is informed of the current directory and is prompted for the name of the file containing the user defined modules:

**Current Directory is :**<br>**D:\CAD\Convrt**

**Input the MODULES.FOR filename:**<br>**Default = MODULE.FOR :**

The user may enter a filename and press <RETURN>, or select the default filename by pressing <RETURN>. If an invalid filename is entered, the user receives the message:

*** * * Invalid Filename * * ***<br>**Do you wish to continue? (Y or N)**<br>**Default = Y:**

If the user selects to continue execution (by entering a “y” and pressing <RETURN>, or by simply pressing <RETURN>), the user is prompted again for the file containing the user defined modules, otherwise program execution is terminated.

Once a valid module file is entered, CONVRT prompts the user for the name of the CADIN file associated with the modules:

**Current Directory is :**<br>**D:\CAD\Convrt**

**Input the CADIN.ASC filename:**<br>**Default = CADIN.ASC :**

The user may enter a filename and press <RETURN>, or select the default filename by pressing <RETURN>. If an invalid filename is entered, the user receives the message:

*** * * Invalid Filename * * ***<br>**Do you wish to continue? (Y or N)**

**Default = Y:**

If the user selects to continue execution (by entering a “y” and pressing <RETURN>, or by simply pressing <RETURN>), the user is prompted again for the CADIN file, otherwise program execution is terminated.

Once a valid CADIN file is entered, CONVRT prompts the user for the name of the HEAD file associated with the modules:

**Current Directory is :**
**D:\CAD\Convrt**

**Input the HEAD.ASC filename:**
**Default = HEAD.ASC :**

The user may enter a filename and press <RETURN>, or select the default filename by pressing <RETURN>. If an invalid filename is entered, the user receives the message:

*** * * Invalid Filename * * ***
**Do you wish to continue? (Y or N)**
**Default = Y:**

If the user selects to continue execution (by entering a “y” and pressing <RETURN>, or by simply pressing <RETURN>), the user is prompted again for the HEAD file, otherwise program execution is terminated.

Once a valid HEAD file is entered, CONVRT prompts the user to select the type of CADRTxx.FOR file to create, MIL Frame or Test Bed:

**Create Real-Time CADAC for**
**MIL Frame (1) or Test Bed (2):**
**Default = MIL Frame (1): 2**

The user may enter 2 if the Test Bed CADRTxx.FOR file is to be created; otherwise the user may enter 1 and press <RETURN> or simply press <RETURN> to have CONVRT create the MIL frame file. Once the user enters a selection, CONVRT prompts the user for the Missile ID:

**Input MISSILE ID -**
**Return for none:**

The user may enter the 2 digit missile ID and press <RETURN> to continue execution or press <RETURN> for no missile ID and continue execution. The missile ID is not a required input, however, for creating multiple real-time CADAC files, the missile ID helps in the identifying each real-time model.

Program execution continues with CONVRT prompting the user for the target initialization file:

**Current Directory is :**
**D:\CAD\Convrt**

**Input the INITIALIZATION filename:**
**Default = INIT.ASC :**

The user may enter a filename and press <RETURN>, or select the default filename by pressing <RETURN>. If an invalid filename is entered, the user receives the message:

*** * * Invalid Filename * * ***
**Do you wish to continue? (Y or N)**
**Default = Y:**

If the user selects to continue execution (by entering a "y" and pressing <RETURN>, or by simply pressing <RETURN>), the user is prompted again for the initialization file, otherwise program execution is terminated. CONVRT then executes to completion without any other interaction with the user.

To execute the CADRTxx.FOR file for Test Bed check-out, the user must compile, link and run the CADRTxx.FOR together with UTL2.FOR. When executing the compiled real-time program, the TRACK file must be in the local directory to provide data link updates to the real-time code. It will write selected parameters and diagnostics to the screen or to the TABOUT.ASC file according to the options selected in the HEAD.ASC file. To compare the trajectories created in the batch mode with the real-time trajectories, the TRAJ.BIN or TRAJ.ASC option in the HEAD.ASC file should be exercised. The smoothing and extrapolation of the track data that arrive at a lower data rate than in the batch mode (every integration step) could cause differences in the trajectories together with the random number selections for the noise models.

## A.2. MIL Frame Version

To generate the real-time missile fly out subroutine CADRTxx.FOR, which can be called from the MIL frame, the CONVRT program is executed with the 'MIL Frame' option. The files that must be present in the local directory are MODULE.FOR, CADIN.ASC, HEAD.ASC and INIT.ASC. The version of CADRTxx.FOR created using this method will not execute by itself; this file is included in the MIL frame along with the CADAC matrix utility routines, UTL2.FOR.

When the program is executed, the user is informed of the current directory and is prompted for the name of the file containing the user defined modules:

**Current Directory is :**
**D:\CAD\Convrt**

**Input the MODULES.FOR filename:**
**Default = MODULE.FOR :**

The user may enter a filename and press <RETURN>, or select the default filename by pressing <RETURN>. If an invalid filename is entered, the user receives the message:

*** * * Invalid Filename * * ***
**Do you wish to continue? (Y or N)**
**Default = Y:**

If the user selects to continue execution (by entering a "y" and pressing <RETURN>, or by simply pressing <RETURN>), the user is prompted again for the file containing the user defined modules, otherwise program execution is terminated.

Once a valid module file is entered, CONVRT prompts the user for the name of the CADIN file associated with the modules:

**Current Directory is :**
**D:\CAD\Convrt**

**Input the CADIN.ASC filename:**
**Default = CADIN.ASC :**

The user may enter a filename and press <RETURN>, or select the default filename by pressing <RETURN>. If an invalid filename is entered, the user receives the message:

*** * * Invalid Filename * * ***
**Do you wish to continue? (Y or N)**
**Default = Y:**

If the user selects to continue execution (by entering a "y" and pressing <RETURN>, or by simply pressing <RETURN>), the user is prompted again for the CADIN file, otherwise program execution is terminated.

Once a valid CADIN file is entered, CONVRT prompts the user for the name of the HEAD file associated with the modules:

**Current Directory is :**
**D:\CAD\Convrt**

**Input the HEAD.ASC filename:**
**Default = HEAD.ASC :**

The user may enter a filename and press <RETURN>, or select the default filename by pressing <RETURN>. If an invalid filename is entered, the user receives the message:

*** * * Invalid Filename * * ***
**Do you wish to continue? (Y or N)**
**Default = Y:**

If the user selects to continue execution (by entering a "y" and pressing <RETURN>, or by simply pressing <RETURN>), the user is prompted again for HEAD file, otherwise program execution is terminated.

Once a valid HEAD file is entered, CONVRT prompts the user to select the type of CADRTxx.FOR file to create, MIL Frame or Test Bed:

**Create Real-Time CADAC for**
**MIL Frame (1) or Test Bed (2):**
**Default = MIL Frame (1):**

The user may enter 2 if the Test Bed CADRTxx.FOR file is to be created; otherwise the user may enter 1 and press <RETURN> or simply press <RETURN> to have CONVRT create the MIL frame file. Once the user enters a selection, CONVRT prompts the user for the Missile ID:

**Input MISSILE ID -**
**Return for none:**

The user may enter the 2 digit missile ID and press <RETURN> to continue execution or press <RETURN> for no missile ID and continue execution. The missile ID is not a required input, however, for creating multiple real-time CADAC files, the missile ID helps in the identifying each real-time model.

Program execution continues with CONVRT prompting the user for the target initialization file:

**Current Directory is :**
**D:\CAD\Convrt**

**Input the INITIALIZATION filename:**
**Default = INIT.ASC :**

The user may enter a filename and press <RETURN>, or select the default filename by pressing <RETURN>. If an invalid filename is entered, the user receives the message:

*** * * Invalid Filename * * ***
**Do you wish to continue? (Y or N)**
**Default = Y:**

If the user selects to continue execution (by entering a "y" and pressing <RETURN>, or by simply pressing <RETURN>), the user is prompted again for the initialization file, otherwise program execution is terminated. CONVRT then executes to completion without any other interaction with the user.