Technical Report

CMU/SEI-87-TR-033
ESD-TR-87-196

Inertial Navigation System Simulator: Behavioral Specification

Stefan F. Landherr
Mark H. Klein
October 1987

Technical Report

CMU/SEI-87-TR-33

ESD-TR-87-196

October 1987

Inertial Navigation System Simulator: Behavioral Specification



Stefan F. Landherr

Resident Affiliate

Mark H. Klein

Ada Embedded Systems Testbed Project

Approved for public release.
Distribution unlimited.

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

This technical report was prepared for the

SEI Joint Program Office
ESD/XRS
Hanscom AFB, MA 01731

The ideas and findings in this report should not be construed as an official DoD position. It is published in the interest of scientific and technical information exchange.

Review and Approval

This report has been reviewed and is approved for publication.

FOR THE COMMANDER

SEI Joint Program Office SIGNATURE ON FILE

This work is sponsored by the U.S. Department of Defense.

This document is available through the Defense Technical Information Center. DTIC provides access to and transfer of scientific and technical information for DoD personnel, DoD contractors and potential contractors, and other U.S. Government agency personnel and their contractors. To obtain a copy, please contact DTIC directly: Defense Technical Information Center, Attn: FDRA, Cameron Station, Alexandria, VA 22304-6145.

Copies of this document are also available through the National Technical Information Services. For information on ordering, please contact NTIS directly: National Technical Information Services, U.S. Department of Commerce, Springfield, VA 22161.

Ada is a registered trademark of the U.S. Department of Defense, Ada Joint Program Office.

Inertial Navigation System Simulator
Behavioral Specification

Abstract: The Ada Embedded Systems Testbed Project at the Software Engineering Institute is specifying and developing a representative real-time application. This document augments an original set of specifications written by a Navy affiliate. The purpose of this behavioral specification is to clarify and augment the original.

1. Introduction

The Inertial Navigation System (INS) Simulator system [Meyers 87a] consists of the INS computer, the external computer (EC), the INS simulator program [Meyers 87b], the external computer program [Meyers 87c], and an operator interface to each.

This document specifies the INS simulator program in terms of its external interfaces and its dynamic behavior. The purpose is to clarify and supplement the functional specification [Meyers 87b].

The document contains five chapters:

1. **Introduction**
2. **Input/Output Interfaces**: specifies the external interfaces of the INS simulator computer in terms of the data structures that are transferred and the layout of the information presented to the operator (i.e., a static view).
3. **External Behavior**: describes the externally visible behavior of the INS simulator program in terms of the responses to specified inputs and the conditions for generating particular outputs (i.e., a dynamic view).
4. **Internal Behavior**: describes those aspects of the behavior of the INS simulator program that are not directly visible (e.g. motion simulation calculations).
5. **Initialization, Control, and Termination**: describes the overall process of initializing, controlling, and terminating the INS simulator program.

Two appendices are included:

A. Timing Constraints: contains a summary of timing constraints that were extracted from the functional specification [Meyers 87b].

B. Communications Link Statecharts: contains a collection of state transition diagrams (statecharts) that define in detail the required behavior of the communications link.

Because it is a requirement to implement the INS simulator program on a variety of computers, some implementation-dependent details have been left unspecified.

2. Input/Output Interfaces

This chapter specifies the external interfaces of the INS simulator computer in terms of the data structures that are transferred and the layout of the information presented to the operator. This is a static view of the external interfaces; the externally visible, dynamic behavior of the INS simulator program is specified in the next chapter.

Figure 2-1 shows a high-level view of the external interfaces of the INS simulator computer. The actual physical interfaces are highly implementation dependent and will not be specified here. Each of the following sections will describe one of these interfaces: communications link between the two computers, interface to the keyboard, interface to the screen, and interface to a disk file.

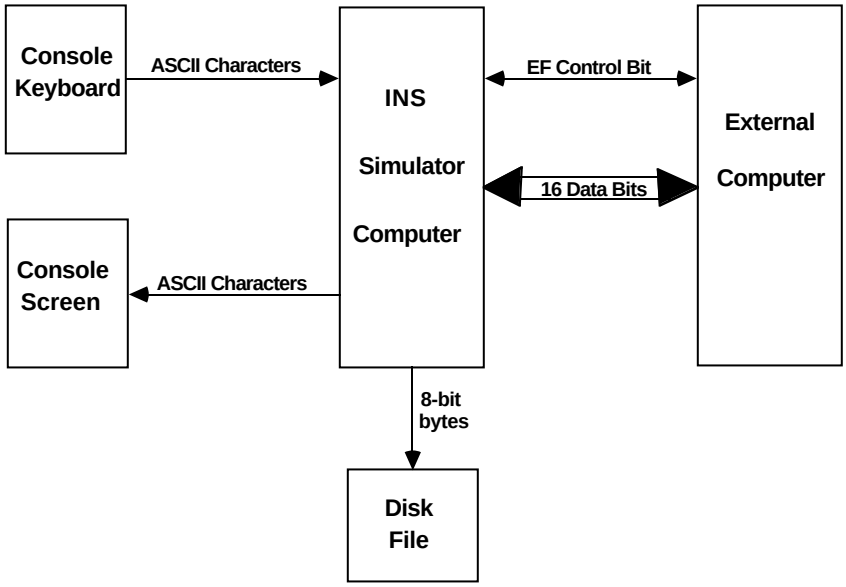


Figure 2-1: Inertial Navigation System Computer Interfaces

2.1. Communications Link

The communications link is used to transfer messages between the INS simulator computer and the external computer system.

2.1.1. Logical Interface

As shown in Figure 2-1, the logical interface to or from the external computer consists of a stream of 17-bit elements, each element consisting of a 16-bit data word and an associated external function (EF) control bit. If the EF bit is high, the data word is interpreted as an external function code; otherwise it is interpreted as a normal message word.

2.1.2. External Function (EF) Codes

The EF codes are used to control the communications protocol and to delimit messages. The code identifiers and their functions are listed in Table 2-1. The actual bit patterns are specified in [NAVSEA 82]. Note that not all the EF codes defined in [NAVSEA 82] are used in the INS simulator application.

Code	Function
ATTN1	Indicate a time-out condition
ATTN2	Enable communications
ATTN4	Disable communications (sent by EC only)
SOTM	Start of test message
SOM	Start of message
RTR	Ready to receive
NRTR	Not ready to receive
EOM	End of message
ACK	Acknowledge (i.e., received a valid message)
NAK	Not-Acknowledge (i.e., received an incomplete or invalid message)

Table 2-1: External Function (EF) Codes

2.1.3. Message Types and Formats

The full range of message types and formats is defined in [NAVSEA 82]. The INS simulator application uses only some of these message types. The message types that may be transmitted to the EC are listed in Table 2-2. The message types that may be received from the EC are listed in Table 2-3. The contents, but not the detailed formats, of these messages are depicted in Tables 2-4, 2-5, 2-6, 2-7, and 2-8. Each message begins with a 2-word header block which specifies the message type and the word count.

Message Type	Message Contents
Test Message	Contains a fixed pattern to allow checking of communications
Time and Status Data Message	Contains fields for the time-of-day and various status codes
Attitude Data Periodic Message	Contains various fields of numerical data pertaining to the (simulated) ship motion
Navigation Data Periodic Message	Contains fields of numerical data pertaining to the (simulated) ship motion

Table 2-2: Messages to EC

Message Type	Message Contents
Test Message	Contains a fixed pattern to allow checking of communications
Select Data Message	Contains fields to select/deselect the periodic messages that may be sent from the INS

Table 2-3: Messages from EC

Message Field	Word Count
Message Header	2 words
Source Identification	1 word
Spare	1 word
Test Word TW1	2 words
Test Word TW0	2 words
---	---
Total	8 words

Table 2-4: Test Message

Message Field	Word Count
Message Header	2 words
Status	2 words
GMT	2 words
Test Word TW1	2 words
Test Word TW0	2 words
---	---
Total	10 words

Table 2-5: Time and Status Data Message

Message Field	Word Count
Message Header	2 words
Data Selection	1 word
Spare	1 word
Test Word TW1	2 words
Test Word TW0	2 words
---	---
Total	8 words

Table 2-6: Select Data Message

Message Field	Word Count
Message Header	2 words
Ownship Heading	1 word
Ownship Pitch	1 word
Ownship Roll	1 word
Ownship Heading Rate	1 word
Ownship Pitch Rate	1 word
Ownship Roll Rate	1 word
GMT	2 words
East Component of Ownship Velocity	1 word
North Component of Ownship Velocity	1 word
Vertical Component of Ownship Velocity	1 word
Ownship Speed	1 word
Test Word TW1	2 words
Test Word TW0	2 words
---	---
Total	18 words

Table 2-7: Attitude Data Periodic Message

Message Field	Word Count
Message Header	2 words
Latitude	2 words
Longitude	2 words
East Component of Ownship Velocity	1 word
North Component of Ownship Velocity	1 word
East Component of Ocean Current	1 word
North Component of Ocean Current	1 word
Ownship Speed	1 word
EM Log Calibration Constant	1 word
Ownship Heading	1 word
Ownship Pitch	1 word
Ownship Roll	1 word
Radial Error Estimate	1 word
Time of Gyro Reset	2 words
GMT	2 words
SOM GMT	2 words
Integral of Velocity North	2 words
Integral of Velocity East	2 words
Test Word TW1	2 words
Test Word TW0	2 words
Total	30 words

Table 2-8: Navigation Data Periodic Message

.

2.2. Console Keyboard

The console keyboard is used to allow the operator to type in various commands.

2.2.1. Logical Interface

As shown in Figure 2-1, the logical interface to the console keyboard consists of a stream of ASCII characters.

The following characters are accepted from the keyboard:

a .. z, **A .. Z**, **0 .. 9**, **=**, **+**, **-**, **.**, **horizontal tab**, **space**, **backspace**, **delete**, **escape**, **carriage return**

All other characters are ignored.

2.2.2. Command Syntax

The functions of the control characters are shown in Table 2-9.

Character	Function
ESC	A special signal to the alerts processing function (see Section 3.3.2)
HT	Equivalent to a space
BS	Used to delete the previous character
DEL	Used to delete the command string
CR	Signals the end of a command string

Table 2-9: Keyboard Control Characters

The non-control characters (including the space character) are used to construct operator commands as specified by the syntax equations in Tables 2-10 and 2-11 on the following page.

SET <parameter-name> = <parameter-value>

SHOW {<parameter-name> | *}

FAULT <variable-name> = <fault-value>

TURN TO {PORT | STARBOARD} AT <turn-rate> UNTIL COURSE <new-course>

{INCREASE | DECREASE} SPEED TO <speed-value> IN <time-period>

RESET GYRO

ENTER

USE FILE <file-name>

{ENABLE | DISABLE} DEX

SELECT {SEASTATE | SCENARIO} <n>

BEGIN

PAUSE

STOP

CLEAR

Notes:

1. <parameter-name> is any input parameter to the motion simulation and <parameter-value> is any legal value for the parameter.
2. <variable-name> is any data variable in an output message to the EC and <fault-value> is any value which can occupy the designated storage allotment for that variable in the output message.
3. <speed-value>/<time-period> must be less than 800 knots per hour.
4. <parameter-name> and range of <parameter-value> must be verified after issuing the SET command (actual value of parameter is not changed until ENTER command is issued).
5. All numeric values are expressed in fixed point notation which accepts signed and unsigned integers and real numbers.
6. The note after Table 3-3 distinguishes between commands that are specified in [Meyers 87b] and those that have been added by the designers.

Table 2-10: Operator Command Syntax, part 1

<parameter-name>	UNITS	MIN*	MAX*
HEAVE_AMP	feet		
HEAVE_FREQ	radians/sec		
HEAVE_PHASE	radians	0	360
LAC_A	feet	-250	250
LAC_B	feet	-25	25
LAC_C	feet	-25	25
LATITUDE	degrees	-90	90
LIST	degrees	-2	2
LONGITUDE	degrees	-180	180
OCEAN_E	knots		
OCEAN_N	knots		
PITCH_AMP	degrees		
PITCH_FREQ	radians/sec		
PITCH_PHASE	radians	0	360
ROLL_AMP	degrees		
ROLL_FREQ	radians/sec		
ROLL_PHASE	radians	0	2*Pi
SHIP_COURSE	degrees	0	360
SHIP_SPEED	knots	0	40
SURGE_AMP	feet		
SURGE_FREQ	radians/sec		
SURGE_PHASE	radians	0	2*Pi
SWAY_AMP	feet		
SWAY_FREQ	radians/sec		
SWAY_PHASE	radians	0	2*Pi
TRIM	degrees	-2	2
YAW_AMP	feet		
YAW_FREQ	radians/sec		
YAW_PHASE	radians	0	2*Pi
<turn-rate>	degrees/sec	0	2
<new-course>	degrees	0	360
<speed-value>	knots	0	40
<time-period>	minutes	0	120
<file-name>	alphanumeric		
<n>	integer		

Table 2-11: Operator Command Syntax, part 2

*Blank spaces indicate that minimums and maximums are to be determined.

2.3. Console Screen

The console screen is used to display some system status indicators and numerical quantities pertaining to the simulated motion of the ship.

2.3.1. Logical Interface

As shown in Figure 2-1, the logical interface to the console screen consists of a stream of ASCII characters. The console screen is assumed to display at least 24 lines of 80 characters. The sequences of control characters required to position the cursor are highly implementation dependent and are not described here.

2.3.2. Screen Layout

The screen is divided into four windows as shown in Figure 2-2. The detailed layout of the the screen is shown in Figure 2-3. Note that the command window, alert window, and the system status window are allocated two lines, but they actually consist of one line of information and a blank line for window separation.

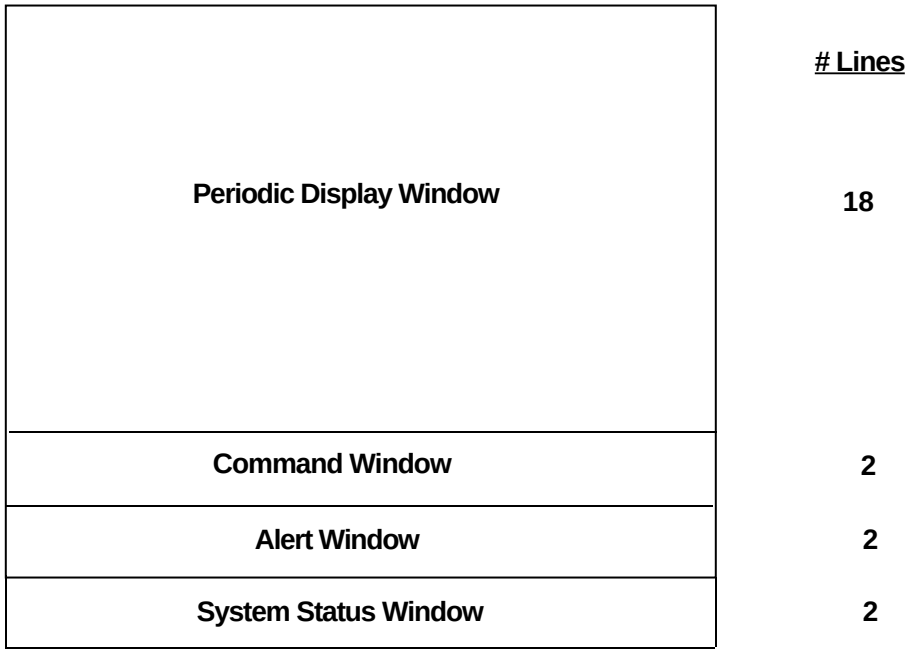


Figure 2-2: Screen Windows

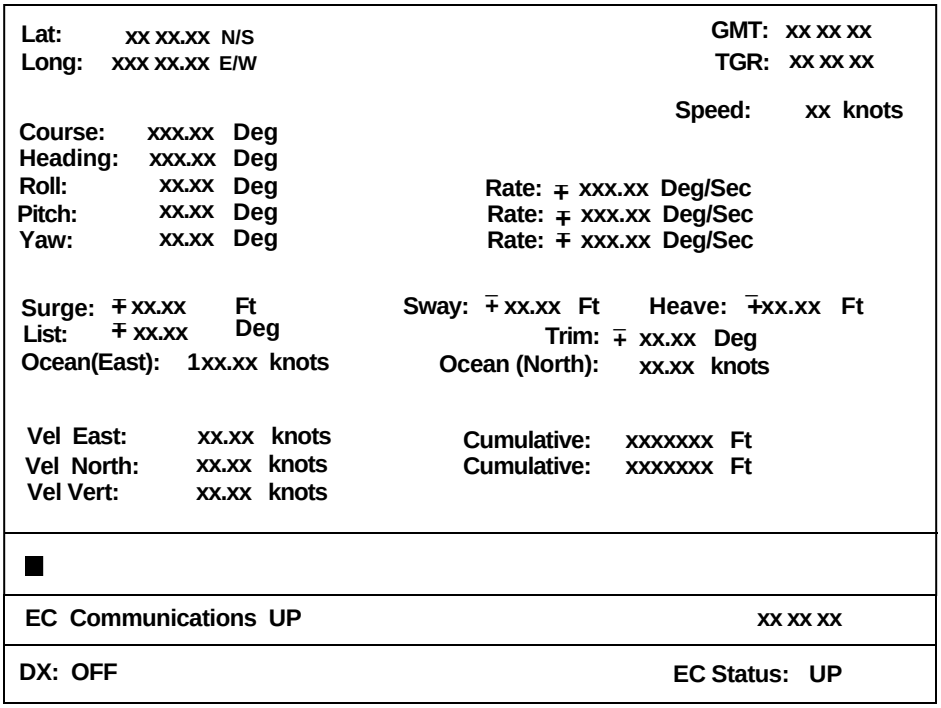


Figure 2-3: Detailed Screen Layout

2.4. Disk File

The disk file is used by the data extraction function to record various data items.

2.4.1. Logical Interface

The logical interface to the disk file consists of a stream of 8-bit bytes, grouped into blocks.

The codes used to control the interaction with the disk file are highly implementation dependent and are not described here.

2.4.2. Recording Format

The general format of a block of recorded data is shown in Table 2-12. Identifying codes are expressed as short mnemonics, and numerical values are expressed in HEXASCII notation. Thus, each block consists of a sequence of printable ASCII characters, terminated by a CR/LF combination.

Field Name	Field Description
event type	two-character code (2 bytes)
timestamp	number of 2.56 msec ticks since program start (8 bytes)
data	character codes or numerical values as appropriate (number of bytes varies with event type)
checksum	modulo 256 checksum (2 bytes)
terminator	ASCII.CR and ASCII.LF characters (2 bytes)

Table 2-12: Data Recording - General Format

The codes for the various event types are shown in Table 2-13.

The timestamp range of 16#00000000# .. 16#FFFFFFFF# ticks is sufficient for several hundred days.

The data recorded for each event type, in addition to the timestamp, are shown in Table 2-14.

The checksum is the modulo 256 sum of all the preceding bytes in the block
(i.e., a number in the range 0..255 decimal, or 16##00#..16##FF#).

Event Code	Event Description
IP	Initialization of sea-state and scenario parameters
CP	Operator command to change a parameter (note: the timestamp is that of the ENTER command)
IF	Operator command to inject a fault
IR	Initiation of runtime BIT processing
IT	Initiation of each runtime test
CT	Completion of each runtime test
CC	Change in state of communications with external computer
IA	Issue of an alert to the operator
RA	Removal of an alert from the alert list
BS	Beginning of session
ES	End of session

Table 2-13: Data Recording - List of Events and Event Codes

Event Code	Data
IP	sea-state number selected (1 byte) scenario number selected (1 byte)
CP	(1) parameter id (4-character code) -- details TBD (4 bytes) (2) new value (in operator input format) (8 bytes)
IF	(1) parameter id (4 character code) -- details TBD (4 bytes) (2) fault value (in operator input format) (8 bytes)
IR	-- no data --
IT	-- no data --
CT	-- no data --
CC	new communications state (4 character code) DOWN down (i.e., disabled or attempting to enable) UP up (i.e., fully enabled)
IA	alert id (4-character code) -- details TBD (4 bytes)
RA	alert id (4-character code) -- details TBD (4 bytes)
BS	date and time (10 bytes for YYMMDDHHMM)
ES	date and time (10 bytes for YYMMDDHHMM)

Table 2-14: Data Recording - Event Data

3. External Behavior

This chapter describes the externally visible behavior of the INS simulator program, i.e., the responses to specified inputs and the conditions for generating particular outputs. The interfaces of concern are:

- interface with the external computer (communications link)
- interface with the operator (keyboard and screen)
- interface to an external medium for data extraction (disk file interface)

3.1. Communications Link

As stated in [Meyers 87b], the communications link between the INS simulator computer and the external computer system must conform to the protocol specified in [NAVSEA 82]. The purpose of this section (and Appendix B) is to give a condensed version of the detailed information in [NAVSEA 82] and in [Meyers 87b].

Communications with the external computer can be in one of three states: disabled, enabling, or enabled. In each of these states, the INS sends and receives data while bound to a specific protocol (i.e., sequence of external function codes and data words). The INS can be viewed as a server to the external computer; that is, the external computer determines INS behavior and can cause preemption of INS message activity. The external computer initiates the enabling and disabling of the INS communications link, directs that certain data be sent or not sent, and periodically requests that the INS respond to test messages.

Sending a successful message consists of an exchange that includes a block of data words, preceded and followed by a pair of EF codes, as detailed in Table 3-1.

Figure 3-1 depicts the overall behavior of the communications link in the ideal case with no transmission errors. The communications link is initially in the disabled state. The external computer initiates the communications protocol with an ATTN2 EF; the INS responds with an ATTN2; and the system enters the enabling state. In the enabling state, the EC sends a test message, and the INS responds with a test message. After the successful exchange of these messages, the system enters the enabled state. In the enabled state, the INS computer accepts and sends messages as dictated by the functional requirements specified in [Meyers 87b] and summarized in Table 3-2. Note that in the case of a conflict, sending an attitude periodic data message takes precedence over sending a navigation periodic data message.

The idealized scenario of Figure 3-1 can be disrupted by a variety of events (e.g., intended recipient not ready, erroneous message, time-out waiting for a response). The full behavior of the communications link is defined in Appendix B.

1. The initiator of a message sends a start-of-message (SOM) or start-of-test-message (SOTM) signal, as appropriate.
2. The recipient, if ready, responds with a ready-to-receive (RTR) signal.
3. The initiator sends the data block, followed by an end-of-message (EOM) signal.
4. If no errors are detected, the recipient responds with an acknowledge (ACK).

Table 3-1: Normal Message Protocol

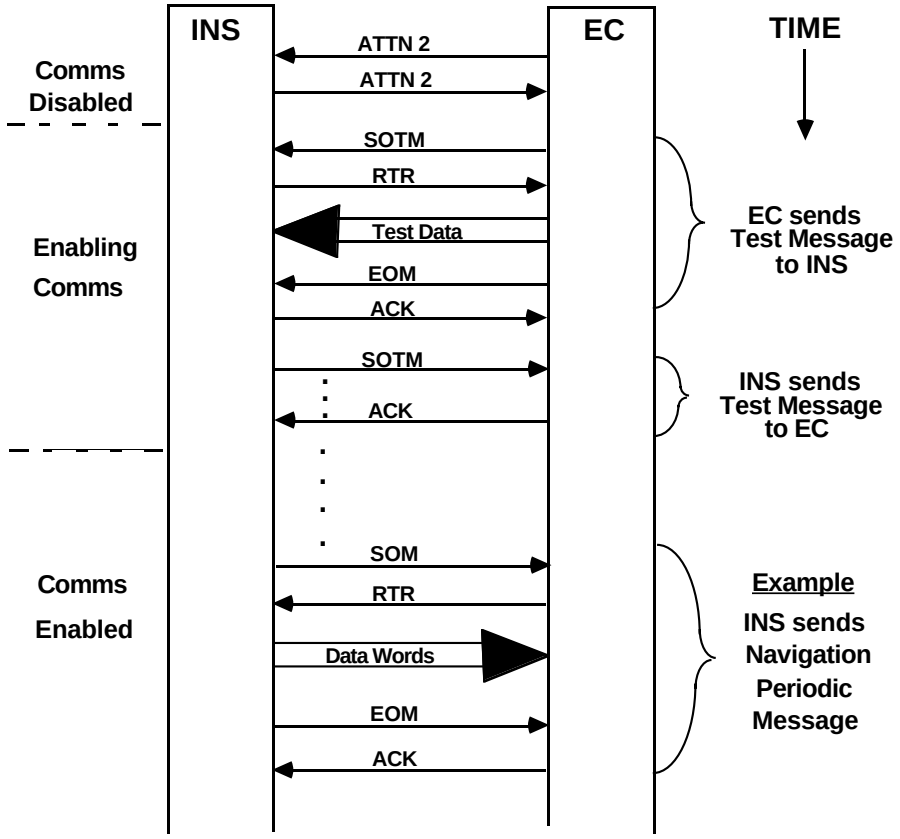


Figure 3-1: Communications Protocol: Summary

Message Type	Conditions
Time and Status Data Message	Sent immediately upon entry to the enabled state, and in response to a select data message from the EC
Test Message	Sent in response to a test message from the EC
Attitude Periodic Message	Sent once every 61.44 seconds, if enabled (by a previous select data message from the EC)
Navigation Periodic Message	Sent once every 983.04 seconds, if enabled

Table 3-2: Conditions for Generating Messages from INS

3.2. Console Keyboard

The effect of each of the operator commands is described in Table 3-3.

Command	Effect
SELECT SEASTATE	Use the specified set of amplitude and frequency parameters for the motion simulation.
SELECT SCENARIO	Use the specified set of ship parameters for the motion simulation.
BEGIN	Start (or restart) the motion simulation.
SET PARAMETER	Save the new value of the specified parameter, to be used upon the next ENTER command.
SHOW PARAMETER	Display the value of the specified simulation parameter.
SHOW *	Display the values of all simulation parameters, usurping the periodic display window of the screen and disabling the normal periodic update of this window.
CLEAR	Erase the periodic display window of the screen, rewrite the fixed legends, and re-enable the normal periodic update of this window.
FAULT	Save the specified fault value of the specified variable; upon the next ENTER command, use that value to inject a fault into the next output data message, in place of its true value.
ENTER	Actually make the changes to the simulation parameters as specified in most recently issued SET PARAMETER and FAULT commands (i.e., issued since the last ENTER command).
change COURSE	Use the new course parameters for the motion simulation.
change SPEED TO	Use the new speed parameters for the motion simulation.
RESET GYRO	Set the Time of Gyro Reset to the current system time.
USE FILE	Open the specified file for use as the data extraction file.
ENABLE DEX	Enable the data extraction function (provided that a file has been specified).
DISABLE DEX	Disable the data extraction function.
PAUSE	Temporarily freeze the simulation (it may be restarted with the BEGIN command).
STOP	Terminate the simulation program.

Table 3-3: Operator Commands

Origin of Commands:

The following commands are specified explicitly in [Meyers 87b]: SET PARAMETER, SHOW PARAMETER, change COURSE, change SPEED,
ENTER, FAULT, RESET GYRO

The following commands are specified implicitly in [Meyers 87b]: SELECT SEASTATE, SELECT SCENARIO, USE FILE, ENABLE DEX, DISABLE DEX

The following commands have been invented to provide needed functionality:

BEGIN, PAUSE, STOP, SHOW *, CLEAR

3.3. Console Screen

The behavior of each window is defined separately.

3.3.1. Command Window

The command line is initially blank. As the operator types a command, the individual (printable) characters are echoed in the command line. If the operator uses the backspace (BS) and delete (DEL) characters to edit the command string, changes are reflected in the command line. The operator indicates the end of a command by typing a carriage-return (CR) character. This CR is not echoed directly; instead a "!" character is appended when a command has been executed successfully, or a "?" character is appended when a command has been found to be invalid.

When a SHOW PARAMETER command is entered, the value of the specified parameter is displayed in the remainder of the command line, in the following format:

SHOW PARAMETER <parameter-name> = <parameter-value> <unit-of-measure> !

where <parameter-name> is as specified in Section 2.2.2,
<parameter-value> is in the appropriate numeric form, and
<unit-of-measure> is as specified in Section 2.2.2.

A command remains on display until the first character of the next command is typed. Thus, the command line displays the following:

1. a blank line
2. an incomplete command string
3. an apparently complete command that has not yet been terminated with a CR
4. a complete command with an indication of whether it has been accepted
5. a SHOW PARAMETER command, followed by the value of the specified parameter

3.3.2. Alert Window

The alert line is blank at program initiation. Immediately after the completion of the Initial Built-In Test, the alert line will display an indication of either a successful BIT or a test failure.

When any of the events listed in Table 3-4 occurs, an alert will be issued (see [Meyers 87b]). If there is no alert currently displayed, the new alert is displayed; otherwise, the new alert is added to a list of pending (capacity of the list is 50 alerts). Additionally, the audible alarm will sound for 2 seconds when an alert is issued.

When the operator types an escape (ESC) character, the currently displayed alert (if any) is erased, and the highest priority pending alert (if any) is removed from the list of pending alerts and displayed. Thus, the alert line is either blank, or it contains the following:

- the alert text string
- the time at which the alert was issued (i.e., detected)

- the number of pending alerts (blank if zero)

RUNTIME BIT FAILURE
INITIAL BIT REGISTER TEST FAILURE
INITIAL BIT ADDRESS READ/WRITE FAILURE
INITIAL BIT ARITHMETIC TEST FAILURE
INITIAL BIT MEMORY CHECKSUM FAILURE
INITIAL BIT TEST SUCCESSFUL
FAULT CHANGES COMPLETED
INVALID MESSAGE TYPE IN MESSAGE
INVALID NUMBER OF WORDS IN MESSAGE
INVALID TEST PATTERN RECEIVED
EC COMMUNICATIONS UP
EC COMMUNICATIONS DOWN
EC COMMUNICATIONS ENABLED
SELECT MESSAGE RECEIVED FROM EC
INVALID COURSE CHANGE
INVALID SPEED CHANGE
INVALID TURN COMMAND
INVALID DX FILE SPECIFIED
UNABLE TO OPEN DX FILE
DX FILE WRITE ERROR
PARAMETER INITIALIZATION COMPLETE
PARAMETER CHANGES COMPLETED
INVALID SET PARAMETER REQUEST
INVALID SHOW PARAMETER REQUEST
INVALID FAULT REQUEST
INVALID DATA EXTRACT REQUEST
INVALID ENTER COMMAND
INVALID ENTRY

NOTES :

1. Alerts are listed in *descending* order of priority.
2. This is the minimal list of alerts specified in [Meyers 87b]. Additional alerts will be defined as required to indicate other erroneous conditions (e.g. time-out detected in the communications link, scheduling deadline missed, buffer overflow).

Table 3-4: List of Alerts

3.3.3. System Status Window

The system status window displays the current status of the communications link (down/up) and the data extraction function (off/on), as shown in Figure 2-3.

The fixed legends are written once, at program initiation, together with the initial values of the status indicators.

When the status of the communications link or the data extraction function changes, the appropriate indicator should change within 1000 milliseconds.

3.3.4. Periodic Display Window

The periodic display window displays various numerical quantities relating to the simulated ship motion, in the format shown in Figure 2-3.

The fixed legends are written once, at program initiation, together with blanks in the numerical fields.

The numerical fields are updated at least once every 1000 milliseconds while the simulation is active (see Chapter 5).

3.4. Disk File Interface (Data Extraction)

The data extraction function is controlled as described in Table 3-3 by the commands:

USE FILE <name>

ENABLE DEX

DISABLE DEX

The data extraction function is initially disabled. When the operator types a USE FILE command, the appropriate disk file is opened (if possible).

When any of the events listed in Table 2-13 occurs and data extraction is enabled, a data extraction record will be written to the disk file. The format of each type of record is specified in Section 2.4.

When the operator terminates the INS simulation program with a STOP command, the disk file is closed.

4. Internal Behavior

This chapter describes the aspects of INS simulator program behavior that are internal to the program, including the motion calculations and the Runtime Built-In Test.

4.1. Motion Calculations

When the motion simulation is active (see Chapter 5), three sets of ship motion calculations are performed at specified frequencies.

4.1.1. Update Ship Attitude

Every 2.56 milliseconds, do the following, as specified in Appendices 2, 3, and 4 of [Meyers 87b]:

- Calculate (simulated) roll and roll rate.
- Calculate (simulated) pitch and pitch rate.
- Calculate (simulated) yaw, heading, and heading rate.

4.1.2. Update Ship Velocity

Every 40.96 milliseconds, do the following, as specified in Appendices 5, 6, 7 and 8 of [Meyers 87b]:

- Update the commanded course if a course change is underway.
- Update the commanded speed if a speed change is underway.
- Calculate surge, heave, and sway.
- Calculate velocity of the ship's center of gravity (CG) with respect to the water.
- Calculate true velocity of the ship's center of gravity.
- Calculate motion at the position of the INS within the ship (attitude and velocity).
- Update the cumulative velocity integrals.

4.1.3. Update Ship Position

Every 1300 milliseconds, do the following, as specified in Appendix 9 of [Meyers 87b]:

- Update the latitude and longitude of the ship.

4.2. Runtime Built-In Test

The Runtime BIT function will be performed every 1000 milliseconds as specified in [Meyers 87b].

The test will determine if the contents of the output message buffers lie within the acceptable bounds specified in [NAVSEA 82].

5. Initialization, Control, and Termination

This chapter describes the process of initializing, controlling, and terminating the INS simulator program.

A typical timeline from program initiation to program termination is shown in Figure 5-1. Note that this timeline represents the ideal case.

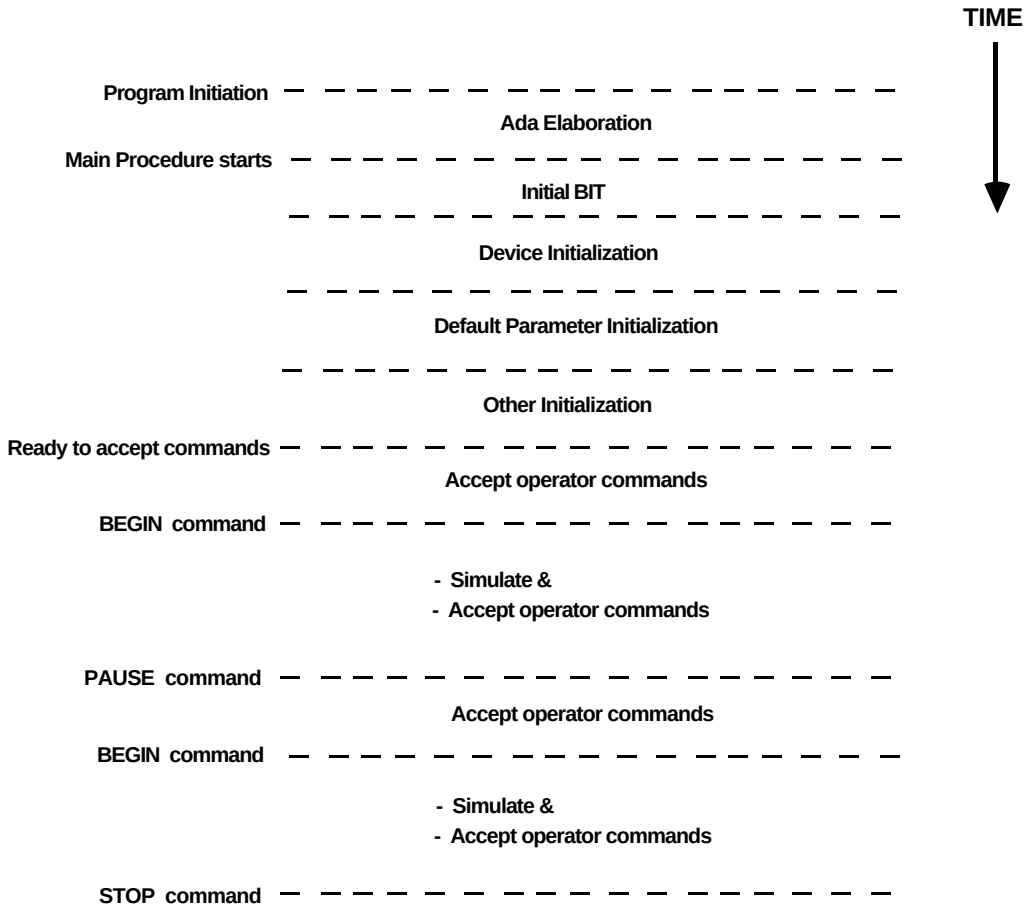


Figure 5-1: Program Timeline

5.1. Program Preparation and Initiation

The process of program preparation and initiation is highly dependent on the host and target systems and will not be specified here.

5.2. Initial Built-In Tests

[Meyers 87b] specifies performance of the following Initial Built-In Tests (see Table 5-1) immediately upon program initiation. However, this poses special problems in Ada because the runtime system performs various package elaborations before transferring control to the main program.

Register checks
Address read/write test
Memory checksum
Arithmetic capability test

Table 5-1: Initial Built-In Tests

Since the first three tests are highly dependent on the implementation system, it may not be feasible to implement them exactly as specified in [Meyers 87b]. However, the arithmetic capability test should be implementable using Ada code only.

5.2.1. Arithmetic Capability Test

This test will check

- the algebraic identity: sqrt(x) * sqrt(x) = x
  for 10 random numbers in the range 10e-10..10e10
- the trigonometric identity: $\sin^2 x + \cos^2 x = 1$
  for 10 random angles in the range 0..2*Pi

The actual random numbers and angles are still to be determined, as are the tolerances. It should also be possible to express the tolerances for these checks in an implementation-independent manner using Ada floating point attributes.

5.3. Program Initialization

After the successful completion of the Initial BIT function, the following program initialization functions are performed in the order given.

5.3.1. Device Initialization

Some implementation-specific device initializations will need to be performed, but they are not described here. Certain implementation-independent initializations will also be performed, as described in Chapter 3.

5.3.2. Motion Simulator Parameter Initialization

A sea-states table contains seven sets of amplitude and frequency parameters to simulate ship's motion in sea-states 1 through 7. A scenarios table contains nine sets of other ship parameters that are required to fully define a simulation. The parameters will be initially set to sea-state 3 and scenario 1.

5.3.3. Other Initialization

The Time of Gyro Reset is set to current wall-clock time.
The state of the communications link is set to DOWN.
The state of the data extraction function is set to OFF.

5.4. Program Control

The program is now ready to accept operator commands from the keyboard and messages from the external computer system. (Any commands or messages received before this point are ignored). The program remains ready to accept operator commands and EC messages until it is terminated. Wholesale re-initialization of the current parameters may be accomplished by these operator commands:

SELECT SEA-STATE <n>

SELECT SCENARIO <n>

Any of the operator commands listed in Table 3-3 may be now be issued.

The simulation starts when the operator enters a BEGIN command.
The simulation is temporarily frozen if the operator enters a PAUSE command; it can be restarted by another BEGIN. The purpose of the PAUSE/BEGIN feature is to assist in debugging and monitoring.)
The program continues until terminated by the operator with a STOP command.

5.5. Program Termination

The INS simulation program is terminated when the operator enters the STOP command, provided that the external computer has already disabled communications. If communications are still enabled, the STOP command is ignored.

Glossary

AEST	Ada Embedded Systems Testbed (Project)
BIT	Built-In Test(s)
CG	center of gravity
DX, DEX	data extraction
EC	external computer system
EF	external function (code)
EM	electro-magnetic
FIFO	first-in first-out
GMT	Greenwich mean time
INS	Inertial Navigation System

References

[Harel 86] Harel, D.
Statecharts: A Visual Formalism for Complex Systems
Science of Computer Programming, 8, 1987
pp. 231-274

[Meyers 87a] Meyers, B. C.
Systems Specification Document for an Inertial Navigation System Simulator and External Computer
Software Engineering Institute, February 1987
To be published.

[Meyers 87b] Meyers, B. C.
Functional Performance Specification for an Inertial Navigation System Simulator
Software Engineering Institute, February 1987
To be published.

[Meyers 87c] Meyers, B. C.
Functional Performance Specification for an External Computer to Interface to an Inertial Navigation System Simulator
Software Engineering Institute, February 1987
To be published.

[NAVSEA 82] NAVSEA
Interface Design Specification for the Inertial Navigation Set AN/WSN-5 to External Computer
NAVSEA T9427-AA-IDS-010/WSN-4, August 1982

Appendix A: Timing Constraints

Time (ms)	Type (*)	Item (**)	Reference (***)
2.56	S	Timestamp for Data Extraction	
2.56	P	Update Ship Attitude	FPS 4.7
	P	Update Ship Heading	FPS 4.9
5.12	T	NRTR "sleep"	IDS 6.3.2.1.b.1
10.24	T	ATTN2 / SOTM Time-Out	IDS 6.2.1.c
10.24	T	SOM / (RTR or NRTR) Time-Out	IDS 6.3.2.1.a
10.24	T	RTR / EOM Time-Out	IDS 6.3.2.2.b
10.24	T	EOM / (ACK or NAK) Time-Out	IDS 6.3.2.1.c
10.24	T	SOTM / (RTR or NRTR) Time-Out	IDS 6.3.2.3.a
40.96	P	Update Ship Speed	FPS 4.6
	P	Update Ship Displacement	FPS 4.8
	P	Update Ship Velocity (& vel integrals)	FPS 4.11
61.44	P	Send Attitude Periodic Message	IDS Table 5-1
983.04	P	Send Navigation Periodic Message	IDS Table 5-1
1000.	P	Perform Runtime BIT	FPS 4.14
1000.	P	Update Status Display on Screen	FPS 4.3.1 (2)
1300.	P	Update Ship Position (Lat & Long)	IDS p 4-11

KEY

(*) Type of Timing Requirement
- P Periodic
- S Timestamp
- T Time-Out

(**) Message EF Codes
- ATTN2 Initialization
- SOTM Start of test message
- SOM Start of message
- EOM End of message
- RTR Ready to receive
- NRTR Not ready to receive
- ACK Acknowledge (i.e., valid message received)
- NAK Not Acknowledge (i.e., invalid message received)

(***) Specification Documents
- IDS Interface Design Specification, AN/WSN-5 to External Computer
- FPS Functional and Performance Specification for INS Simulator

Table A-1: INS Simulator Program: Timing Constraints

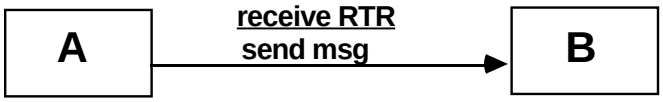
Appendix B: Communications Link Statecharts

This appendix contains a set of statecharts [Harel 86] that describes the behavior of the communications link from the perspective of the INS. This behavior is presented textually in Chapter 6 of [NAVSEA 82]. The goal here is to formalize and clarify.

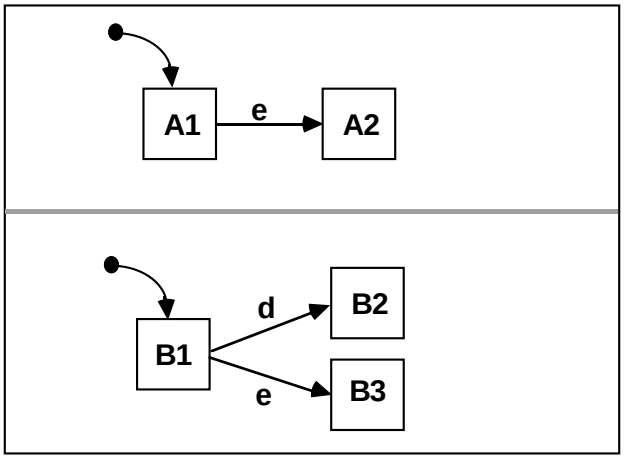
Statecharts incorporate extensions to traditional state transition diagrams that allow for the representation of concurrent states and nested states. The first section below summarizes statechart graphical syntax. The following sections exhibit statecharts with accompanying narrative.

B.a. Summary of Statechart Syntax

1. **States** are represented as boxes. Boxes may be nested, allowing one to view states at varying levels of abstraction.
2. **Transitions** are represented by arrows emanating from a box. Arrows emanating from an outer box represent a transition from any box which it encapsulates. Transitions from several sources may converge on a dot, which also has exiting transitions. This provides an economical mechanism for applying additional conditions and actions to all transitions that converge on the dot.
3. **Events** cause state transitions to take place. They are denoted as labels of a transition.
4. **Actions** may be associated with an event. When actions are present, they appear below a line in the label, where the triggering event appears above the line. An ampersand (**&**) is a separator between multiple actions.



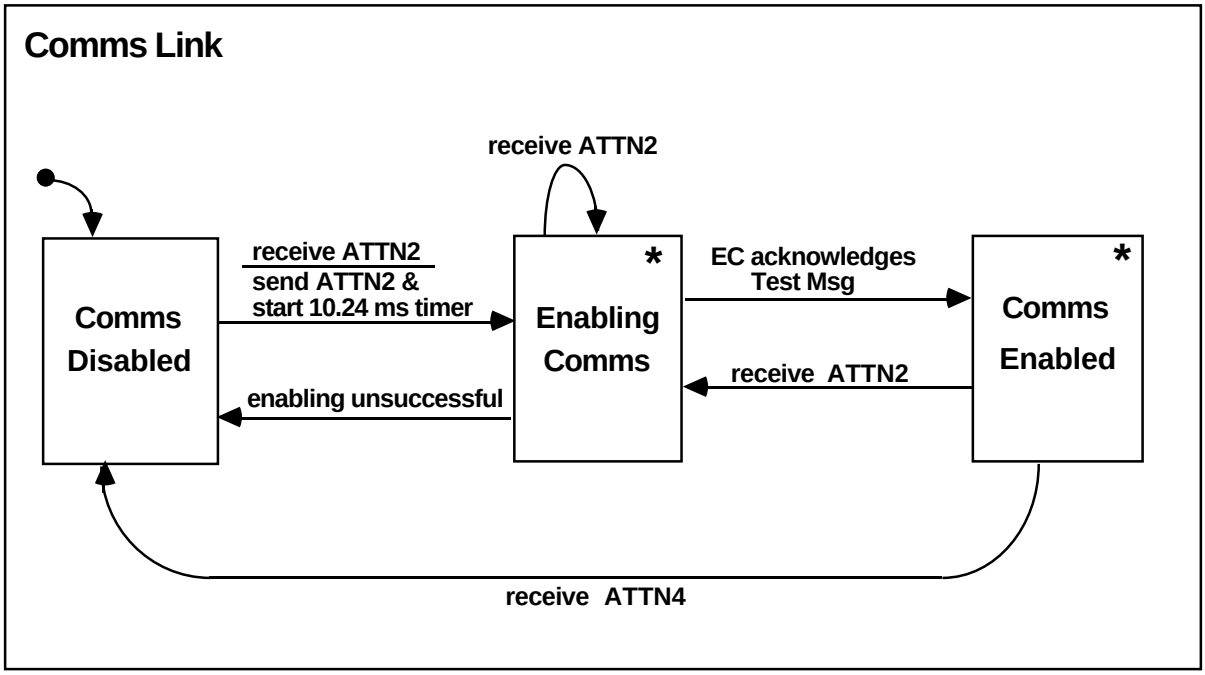
5. **Concurrent states** are represented as two boxes with a common side that is a dotted line.
6. **Initial states** entered when entering a set of encapsulated states are indicated by an arrow with a dot at its tail. In the example below, states ***A1*** and ***B1*** are the initial states that are entered simultaneously, and event ***e*** causes a transition to states ***A2*** and ***B3***.



7. **Conditions** are denoted by text in parentheses. State transitions can be triggered by a true condition.
8. **History** is shown by an arrow that points to an encircled ***H***; the ***H*** indicates that the transition should be made to the most recently exited state.
9. **Expansion** is shown by boxes with an asterisk in the upper right corner; these represent states that have internal detail which is presented in a subsequent statechart.

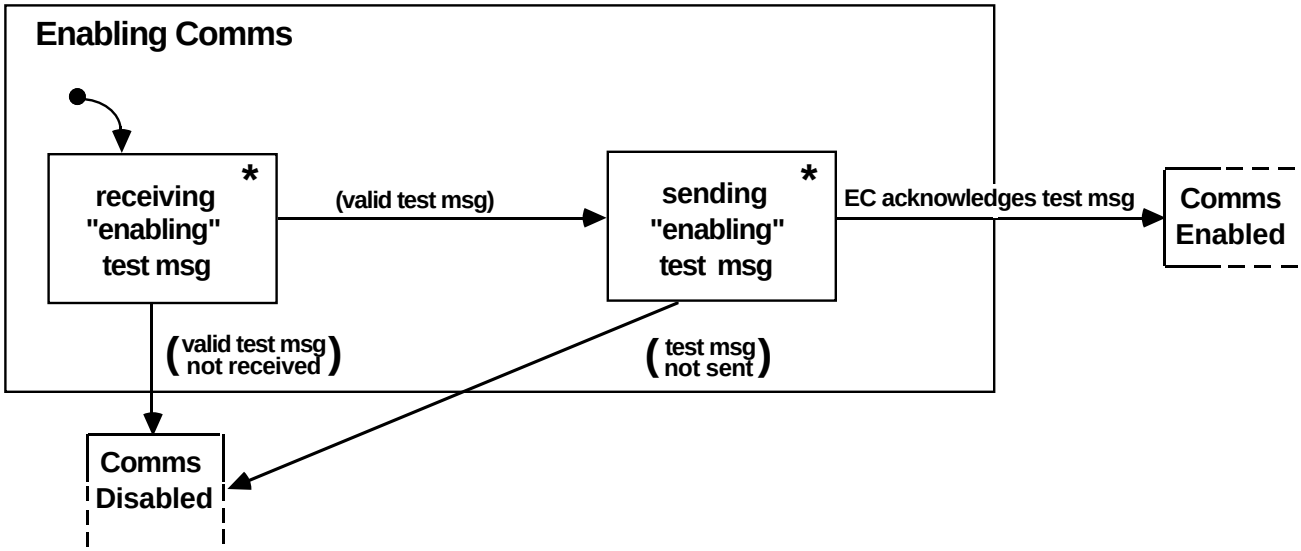
B.b. Master Communications Link Statechart

This chart is the highest level statechart of the INS communications link. It shows the three major states of the communications link: disabled, enabling, and enabled. Receiving an ATTN2 from the external computer precipitates several actions and causes transition to the enabling state. Success in the enabling state results in a transition to the enabled state, and failure to enable results in a transition back to disabled. Note that receiving an ATTN2 and ATTN4 will precipitate the indicated transition from any substate hidden inside the indicated states. Also note that the asterisks in the upper right corner of the boxes indicate that subsequent statecharts exist to show the detail that is hidden at the current level.



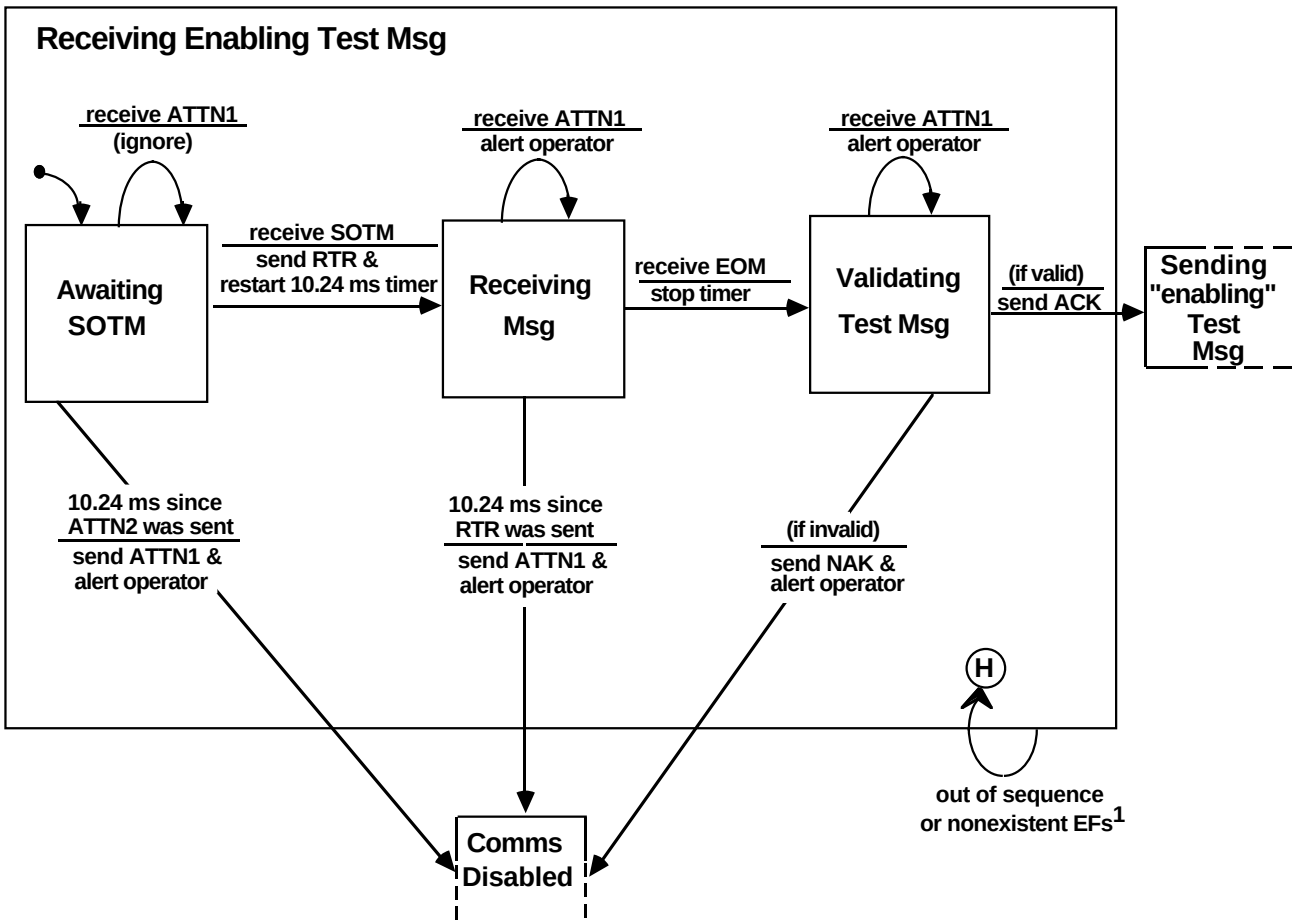
B.c. Enabling Communications Statechart

This statechart is an expansion of the enabling communications state of the higher level. The enabling protocol consists of receiving and sending a test message. Note the use of partial boxes to indicate a state at a higher level.



B.d. Receiving Enabling Test Message Statechart

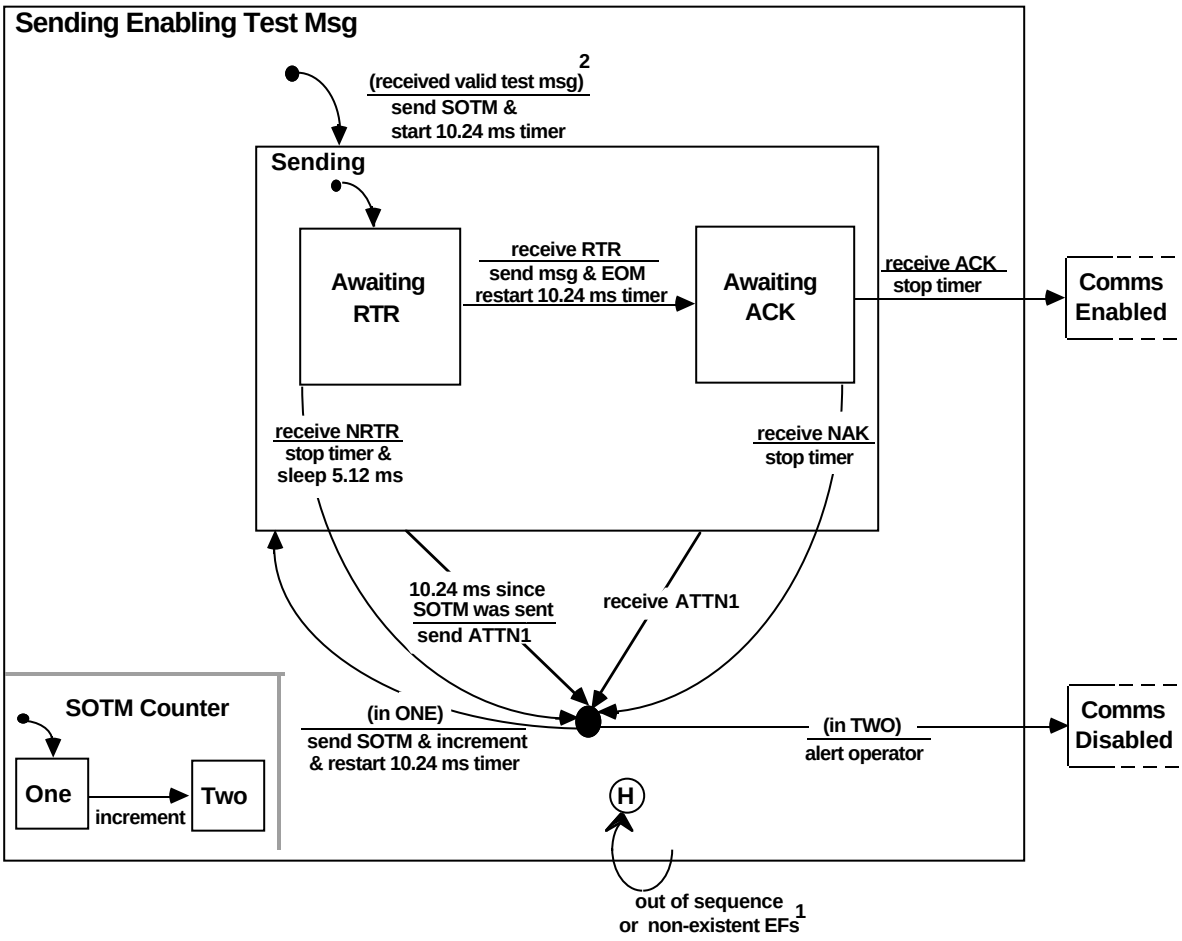
This statechart exhibits the details of receiving a test message during the enabling process. Note that time-outs result in a transition back to the disabled state. Also, receiving an NAK when validating the test message results in a transition to the disabled state. Out of sequence or nonexistent EFs are ignored as indicated by the transition to the encircled **H**, with ATTN2 being the exception.



1 except ATTN2 : see Comms Link statecharts

B.e. Sending Enabling Test Message Statechart

This statechart exhibits the details of sending a test message during the enabling process. Note that the starting transition in this statechart is labeled with an event that also appears on a higher level statechart (see Section B.c). The actions associated with this event are considered unnecessary detail at the higher level and thus are represented at this level. Also note the use of concurrent states to remember the number of attempts that have been made to send the message.



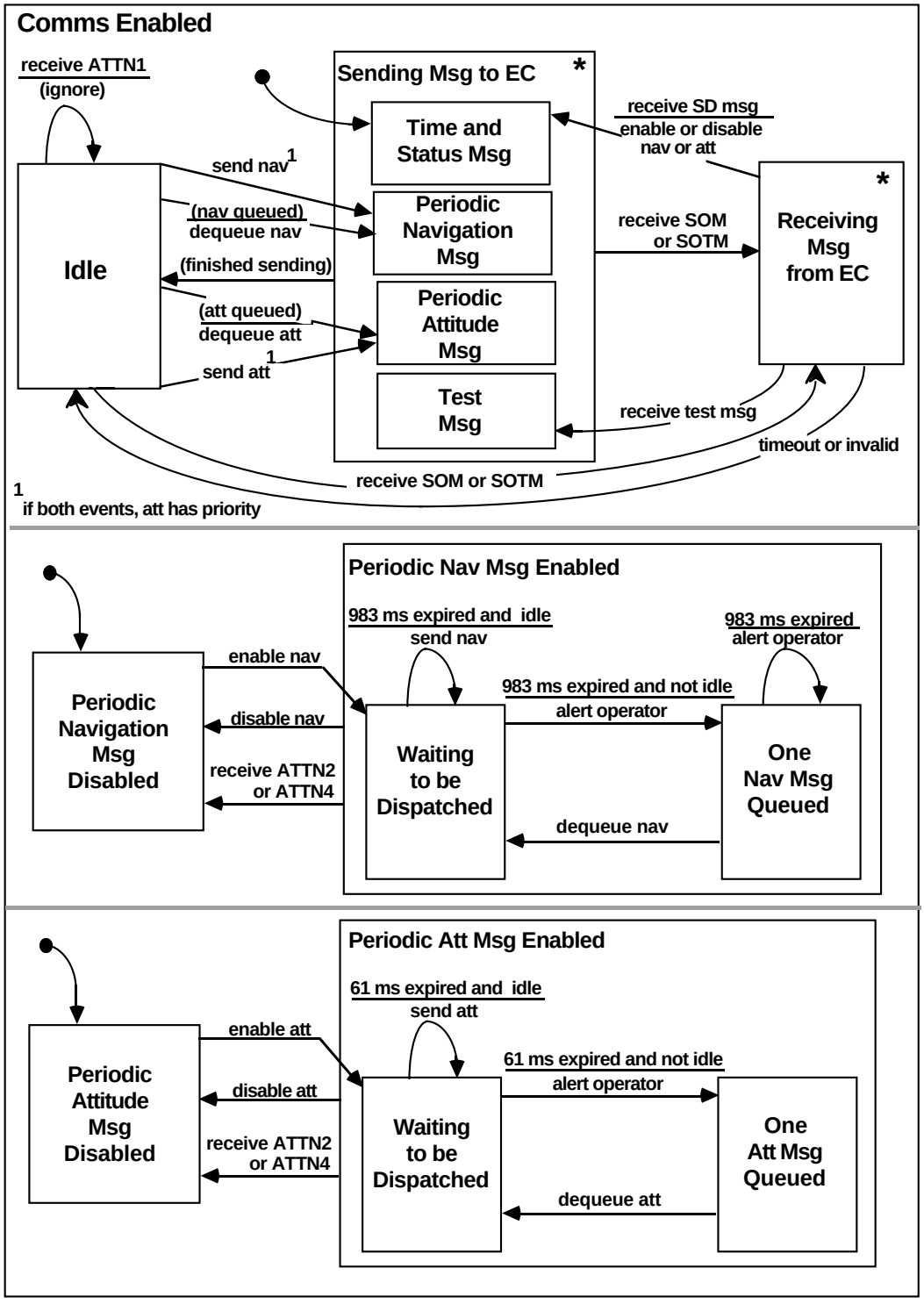
[1] except ATTN2: see higher level statecharts

[2] this event is also shown on previous statechart as "(if valid)"

B.f. Communications Enabled Statechart

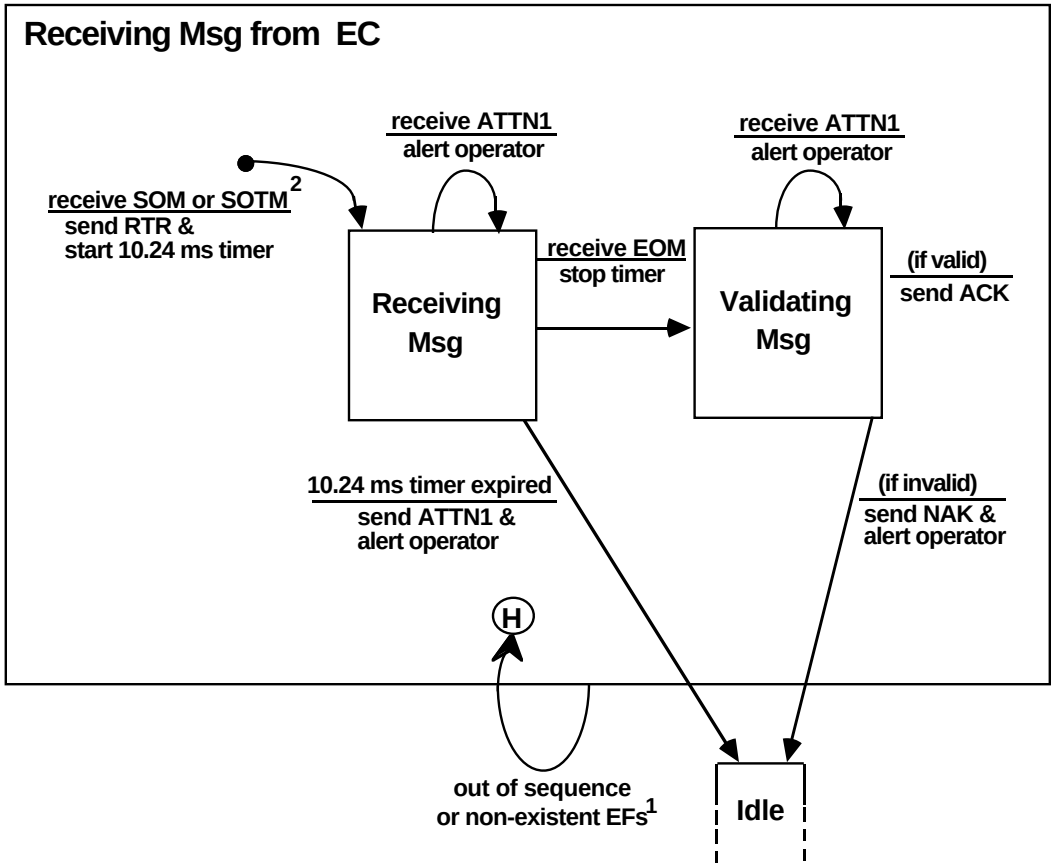
The statechart on the following page presents the details of the communications enabled state as a set of three concurrent states. Upon entering the communications enabled state, the INS sends the time and status message; the sending of the two periodic messages is in a disabled state.

The state, sending message to EC, has four substates, one for each type of message. The detailed statechart for sending each of these messages is common and is exhibited in a later statechart labeled *Sending message to EC* (see Section B.h). Note that if the INS is in the middle of sending a message and either an SOM or SOTM arrive, the original message is aborted and the protocol for receiving a message is enforced. Also notice the interactions between several concurrent states. For example, when the select data (SD) message arrives and requests that the INS send periodic navigation data, the associated action is *enable nav*. This action is also an event which triggers the transition to the state of waiting to be dispatched.


Comms Enabled
receive ATTN1
(ignore)
Sending Msg to EC
*
Time and
Status Msg
receive SD msg
enable or disable
nav or att
send nav 1
(nav queued)
dequeue nav
Periodic
Navigation
Msg
receive SOM
or SOTM
Receiving
Msg
from EC
*
Idle
(finished sending)
(att queued)
dequeue att
1
send att
Periodic
Attitude
Msg
Test
Msg
receive test msg
timeout or invalid
receive SOM or SOTM
1
if both events, att has priority
Periodic Nav Msg Enabled
983 ms expired and idle
send nav
983 ms expired
alert operator
enable nav
983 ms expired and not idle
alert operator
Periodic
Navigation
Msg
Disabled
disable nav
Waiting
to be
Dispatched
One
Nav Msg
Queued
receive ATTN2
or ATTN4
dequeue nav
Periodic Att Msg Enabled
61 ms expired and idle
send att
enable att
61 ms expired and not idle
alert operator
Periodic
Attitude
Msg
Disabled
disable att
Waiting
to be
Dispatched
One
Att Msg
Queued
receive ATTN2
or ATTN4
dequeue att

B.g. Receiving Message from External Computer Statechart

This statechart represents the details for receiving any message when communications is in the enabled state. It is similar to the statechart that shows receiving a test message during enabling (refer to the statechart in Section B.d).

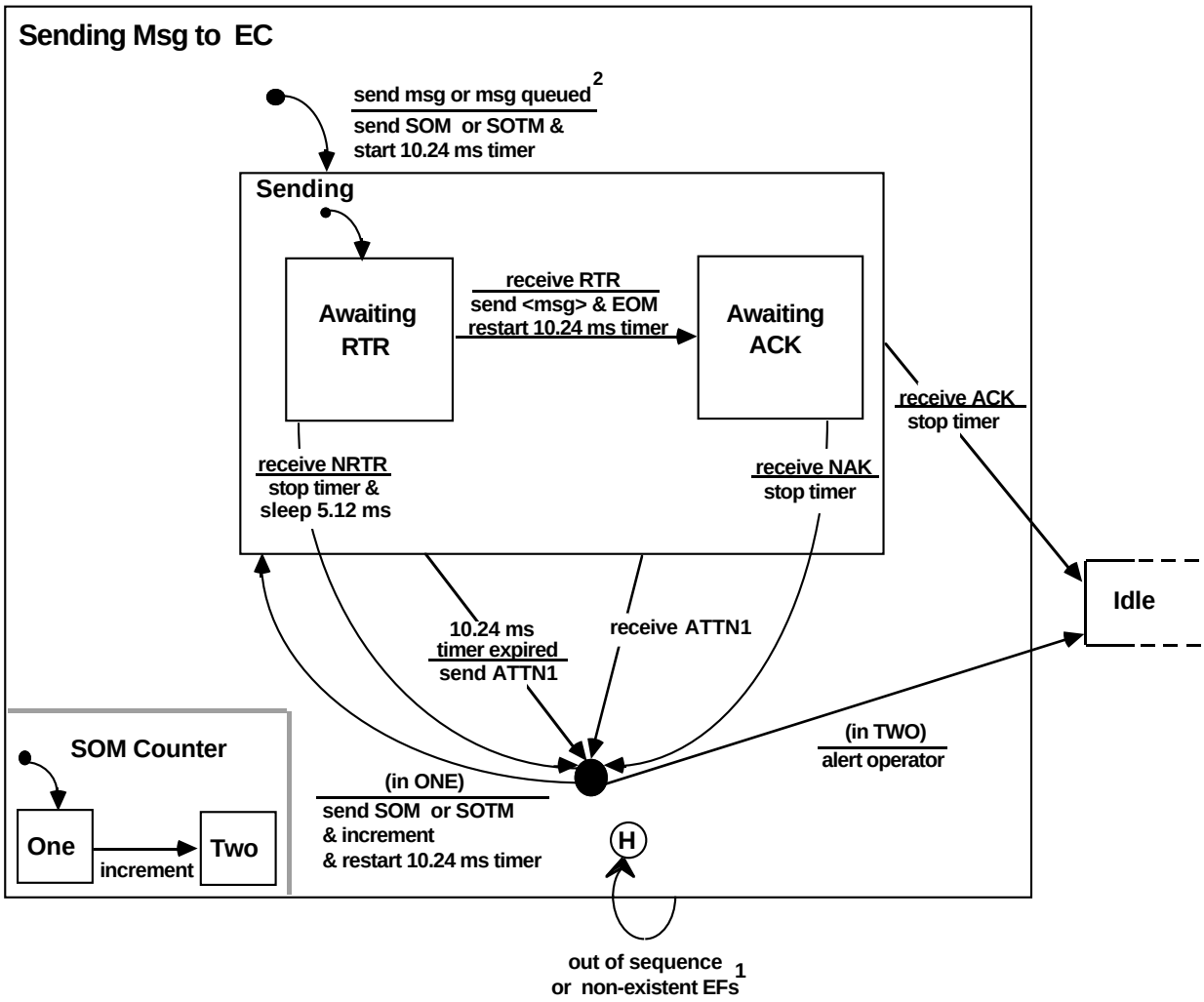


[1] except SOM, SOTM, ATTN2 and ATTN4; see higher level statecharts

[2] this event is also shown on previous statechart

B.h. Sending Message to External Computer Statechart

This statechart represents the details for sending any message when communications is in the enabled state. It is very similar to the statechart that shows sending a test message during enabling (refer to the statechart in Section B.e).



1 except SOM or SOTM and ATTN2 & ATTN4: see higher level statecharts

2 the Comms Enabled statechart also represents this event

Table of Contents

1. Introduction **1**

2. Input/Output Interfaces **3**

2.1. Communications Link 3

2.1.1. Logical Interface 3

2.1.2. External Function (EF) Codes 4

2.1.3. Message Types and Formats 4

2.2. Console Keyboard 8

2.2.1. Logical Interface 8

2.2.2. Command Syntax 8

2.3. Console Screen 11

2.3.1. Logical Interface 11

2.3.2. Screen Layout 11

2.4. Disk File 12

2.4.1. Logical Interface 12

2.4.2. Recording Format 12

3. External Behavior **17**

3.1. Communications Link 17

3.2. Console Keyboard 20

3.3. Console Screen 21

3.3.1. Command Window 21

3.3.2. Alert Window 21

3.3.3. System Status Window 24

3.3.4. Periodic Display Window 24

3.4. Disk File Interface (Data Extraction) 24

4. Internal Behavior **25**

4.1. Motion Calculations 25

4.1.1. Update Ship Attitude 25

4.1.2. Update Ship Velocity 25

4.1.3. Update Ship Position 25

4.2. Runtime Built-In Test 25

5. Initialization, Control, and Termination **27**

5.1. Program Preparation and Initiation 28

5.2. Initial Built-In Tests 28

5.2.1. Arithmetic Capability Test 28

5.3. Program Initialization 29

5.3.1. Device Initialization 29

5.3.2. Motion Simulator Parameter Initialization 29

5.3.3. Other Initialization 29

5.4. Program Control 29

5.5. Program Termination 30

Glossary **31**

References **33**

Appendix A. Timing Constraints **35**

Appendix B. Communications Link Statecharts **37**
B.a. Summary of Statechart Syntax 37
B.b. Master Communications Link Statechart 39
B.c. Enabling Communications Statechart 40
B.d. Receiving Enabling Test Message Statechart 41
B.e. Sending Enabling Test Message Statechart 42
B.f. Communications Enabled Statechart 43
B.g. Receiving Message from External Computer Statechart 45
B.h. Sending Message to External Computer Statechart 46

List of Figures

Figure 2-1: Inertial Navigation System Computer Interfaces 3
Figure 2-2: Screen Windows 11
Figure 2-3: Detailed Screen Layout 12
Figure 3-1: Communications Protocol: Summary 18
Figure 5-1: Program Timeline 27

List of Tables

Table 2-1: External Function (EF) Codes 4
Table 2-2: Messages to EC 4
Table 2-3: Messages from EC 5
Table 2-4: Test Message 5
Table 2-5: Time and Status Data Message 5
Table 2-6: Select Data Message 6
Table 2-7: Attitude Data Periodic Message 6
Table 2-8: Navigation Data Periodic Message 7
Table 2-9: Keyboard Control Characters 8
Table 2-10: Operator Command Syntax, part 1 9
Table 2-11: Operator Command Syntax, part 2 10
Table 2-12: Data Recording - General Format 13
Table 2-13: Data Recording - List of Events and Event Codes 14
Table 2-14: Data Recording - Event Data 15
Table 3-1: Normal Message Protocol 18
Table 3-2: Conditions for Generating Messages from INS 19
Table 3-3: Operator Commands 20
Table 3-4: List of Alerts 23
Table 5-1: Initial Built-In Tests 28
Table A-1: INS Simulator Program: Timing Constraints 35