

# DERIVED PARAMETER IMPLEMENTATION IN A TELEMETRY PREPROCESSOR

Kathleen B. Bossert  
Principal Software Engineer  
Loral Data Systems  
Sarasota, Florida 34230

## KEY WORDS

Derived parameter(s), Telemetry preprocessor

## ABSTRACT

Today's telemetry preprocessing systems are often required to create and process new telemetry parameters by combining multiple actual parameters in a telemetry data stream. The newly created parameters are commonly referred to as "derived parameters" and are often required for analysis in real time at relatively high speeds. Derived parameters are created through algebraic or logical combinations of multiple parameters distributed throughout the telemetry data frame. Creation and processing of derived parameters is frequently performed in telemetry system preprocessors, which are much more efficient at processing time division multiplex data streams than general purpose processors.

Providing telemetry system users with a "user friendly" method for creating and installing newly derived parameter functions has been a subject of considerable discussion. Successful implementation of derived parameter processing has typically required the telemetry system user to be knowledgeable of the telemetry preprocessor architecture and to possess software programming skills.

An innovative technique which requires no programming language skills is presented in this paper. Programmers or non-programmers may use the technique to easily define derived parameter calculations. Both single derived parameters and multiple derived parameters may be calculated in the preprocessor at high throughput rates.

## INTRODUCTION

In the past, setup for a preprocessor was too complex for untrained people to understand. Users had to know preprocessor microcode or another programming language to create

in-box calculated or derived parameters. This included parameters that were created within a preprocessor from one, two, or more input data parameters or measurands. Valuable time was spent learning the programming language and then writing, installing and debugging the algorithm.

If the user did not wish to write preprocessor microcode, then assembly language, and in some cases high level language programs, were written and installed to run on the preprocessor. Normally, this method still required a programmer to create and install the calculation program. In addition, preprocessor memories limited the number of available algorithms that could be maintained.

As another alternative method of data calculation, high level language data analysis programs were written on the host. Again valuable time was spent writing and debugging the program; consequently the efficiency of the high speed preprocessor was lost. By performing the calculations at the host, these real time calculated values were not stored on the archive file during data acquisition, and therefore required recalculation during data playback.

## DERIVED PARAMETER PROCESSING SETUP

Engineers at Loral Data Systems have developed a new method for data calculation and preprocessor setup. The Loral Data Systems EMR O/S90 System allows managers, project engineers and any non-programmer system users to create calculated parameters using simple English language.

A novice user is able to define the parameter equation very similarly to the way he would write an equation on paper. By simply selecting mathematical functions from a pull-down menu list and stringing together several functions, the user may perform such calculations as velocity, engine torque, calibrated altitude, and calculated airspeed.

The O/S90 is based on an open architecture EMR 8715 Telemetry Preprocessor. Although this preprocessor has been in use in the field for many years, open architecture and derived parameter algorithm packaging are new additions. Refer to the overview diagram shown in Figure 1.

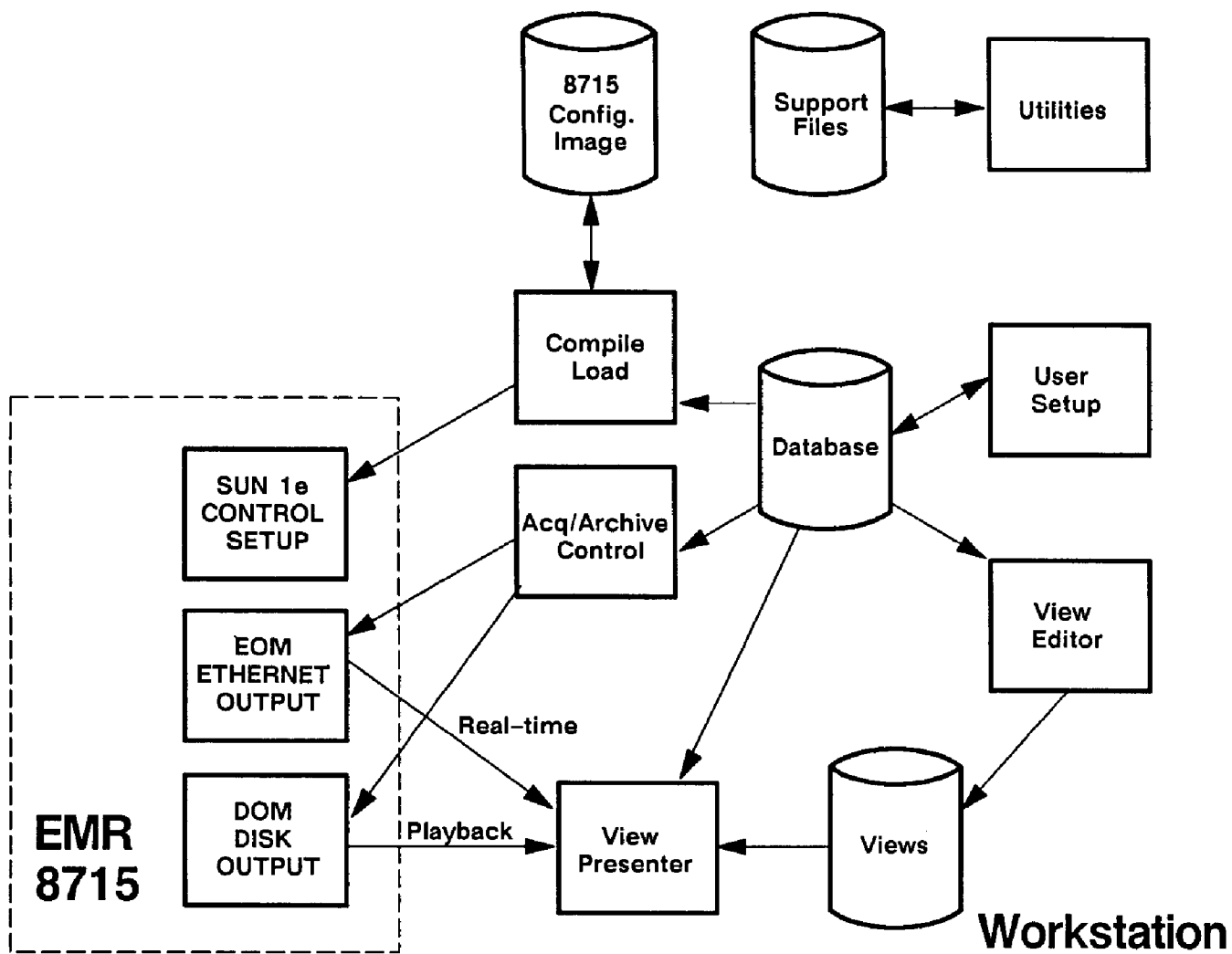


Figure 1. EMR O/S90 System Overview Diagram

The Telemetry Preprocessor is driven via user setup, which resides in the Parameter Database. The Parameter Database Editor is used to define the incoming data stream, data compression and preprocessing to be performed, and output destination for telemetry data. Refer to the Parameter Editor Menu shown in Figure 2.

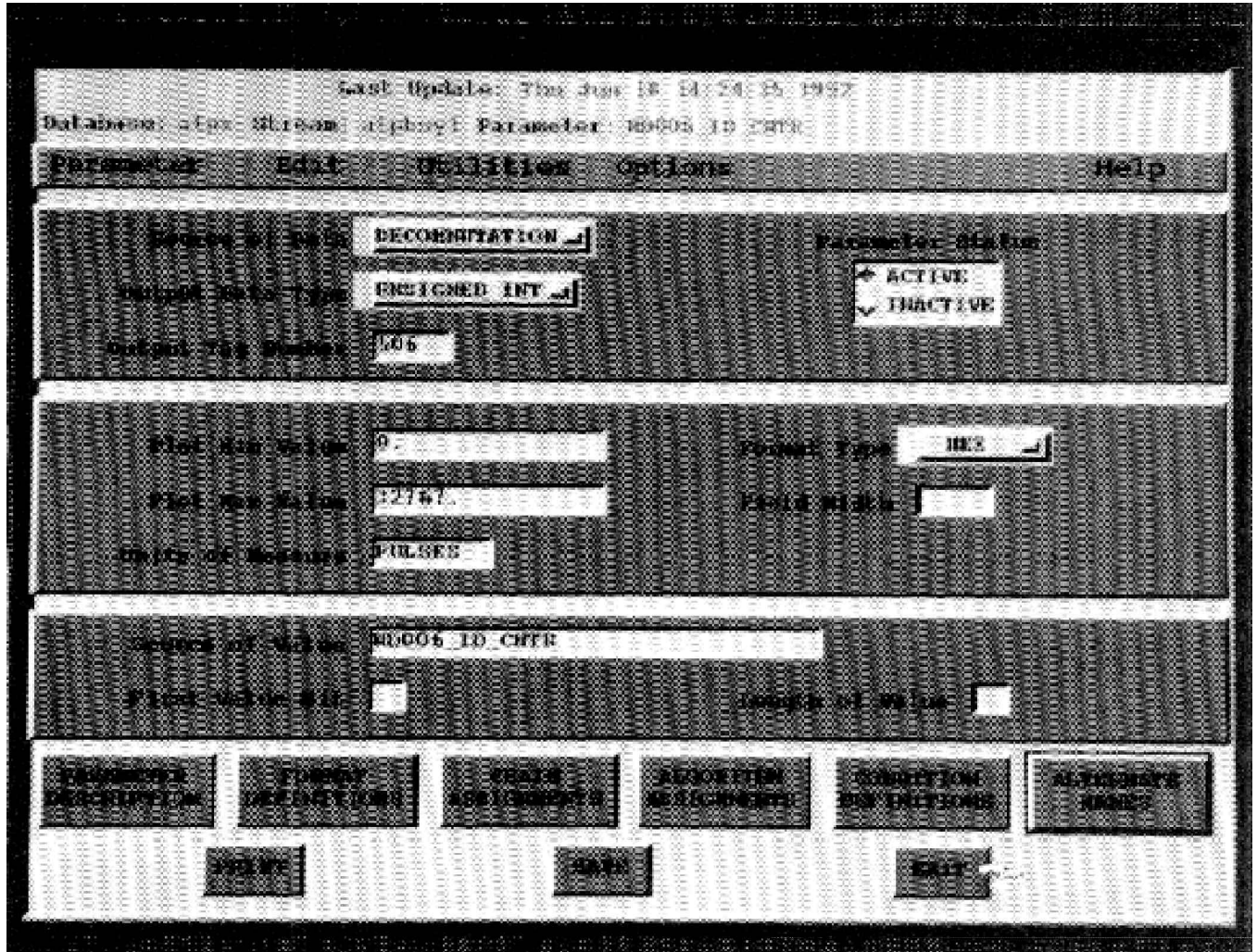


Figure 2. Parameter Editor Menu

The Parameter Editor Format Definitions Menu defines real time input parameters by format location (such as word number, frame number, subcom unit, subframe number, interval). Refer to the Format Definitions Menu in Figure 3.

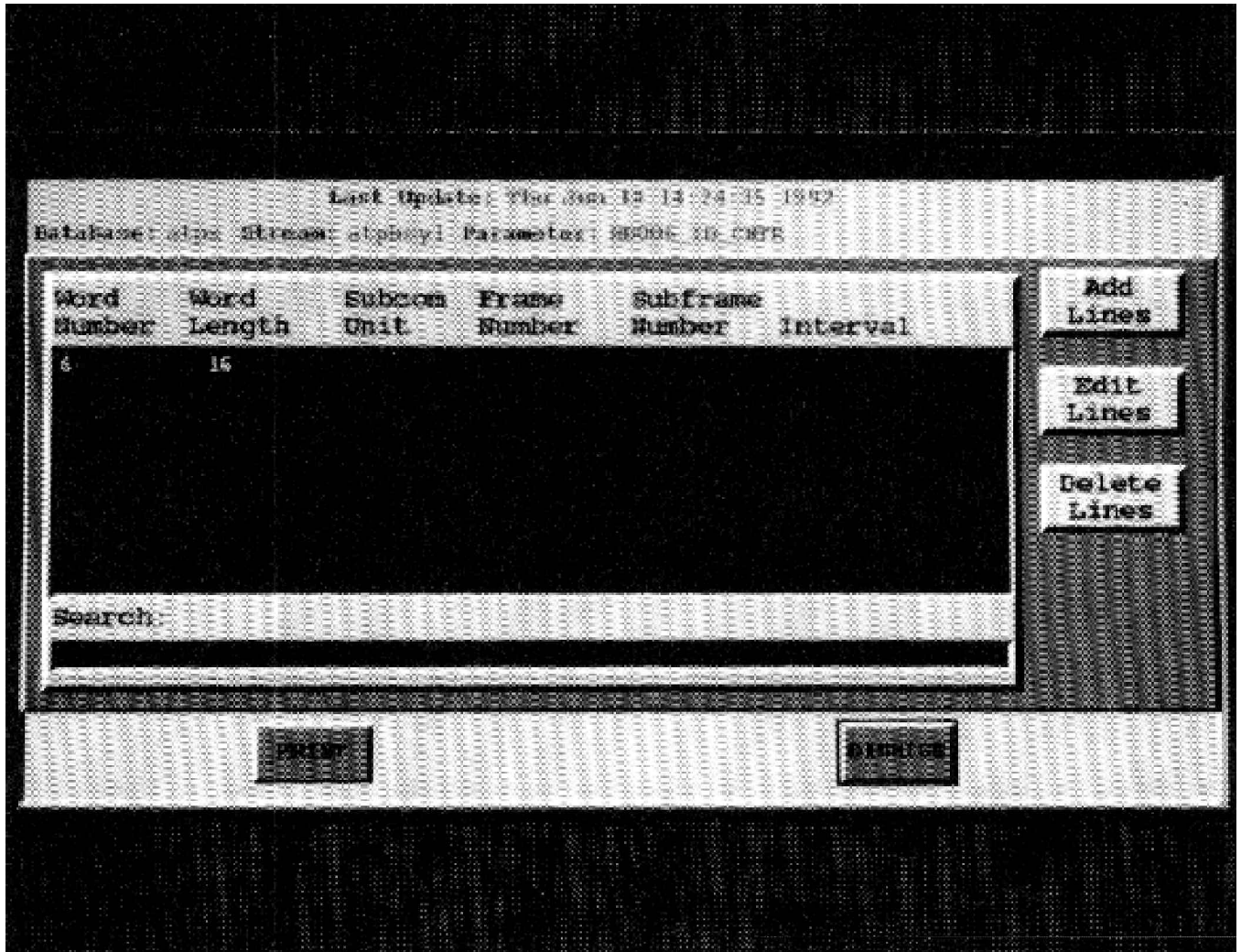


Figure 3. Format Definitions Menu

A series of algorithms, which are selected from a menu list, are assigned an algorithm chain name. The user may assign this predefined chain of algorithms to any of the parameters in the database. The algorithms contained in the original chain may be modified for every parameter in the database as required. Chaining commonly used algorithms and assigning a chain name eliminates the need to redefine the algorithm chain for each new parameter.

Derived parameter algorithms might include any of the normal data compression algorithms, as well as any of the following example set of predefined algorithms:

Addition	Arc Tangent
Subtraction	Arc Cosine
Division	1750 Floating Point
Multiplication	Store Constant to Label
Concatenation	Store Floating Pt. to Label
Arc Sine	Output Label value
Square root	Exponential
Bitwise AND	Bitwise complement (NOT)
Bitwise OR	Bitwise left shift
Bitwise right shift	Natural and common logarithms
Compare equal	Compare not equal
Absolute value	

Data calculation is made simple by creating and storing chains of algorithms in the Parameter Editor and then assigning selected chains to parameter names defined in the database. Figure 4 shows the Parameter Editor Chain Assignments menu, from which the user defines the algorithm chain(s) to be performed during creation of the resultant output parameter.

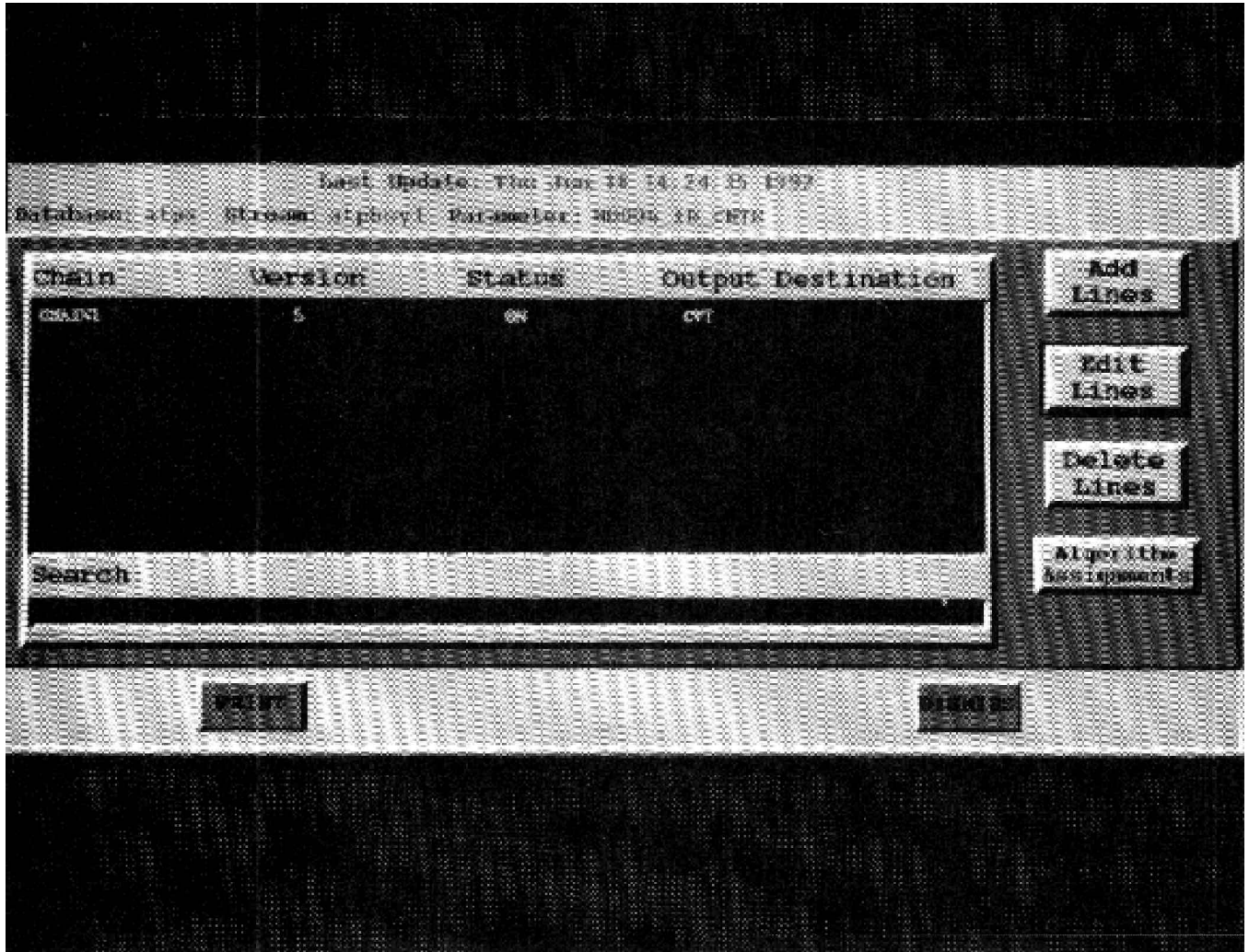


Figure 4. Parameter Editor Chain Assignments Menu



Arguments for algorithms are defined from the Parameter Editor Algorithm Assignments Menu. Information such as input variables, and preprocessor output port selections are defined for each parameter. Refer to Figure 5 for an example of the Algorithm Assignments Menu.

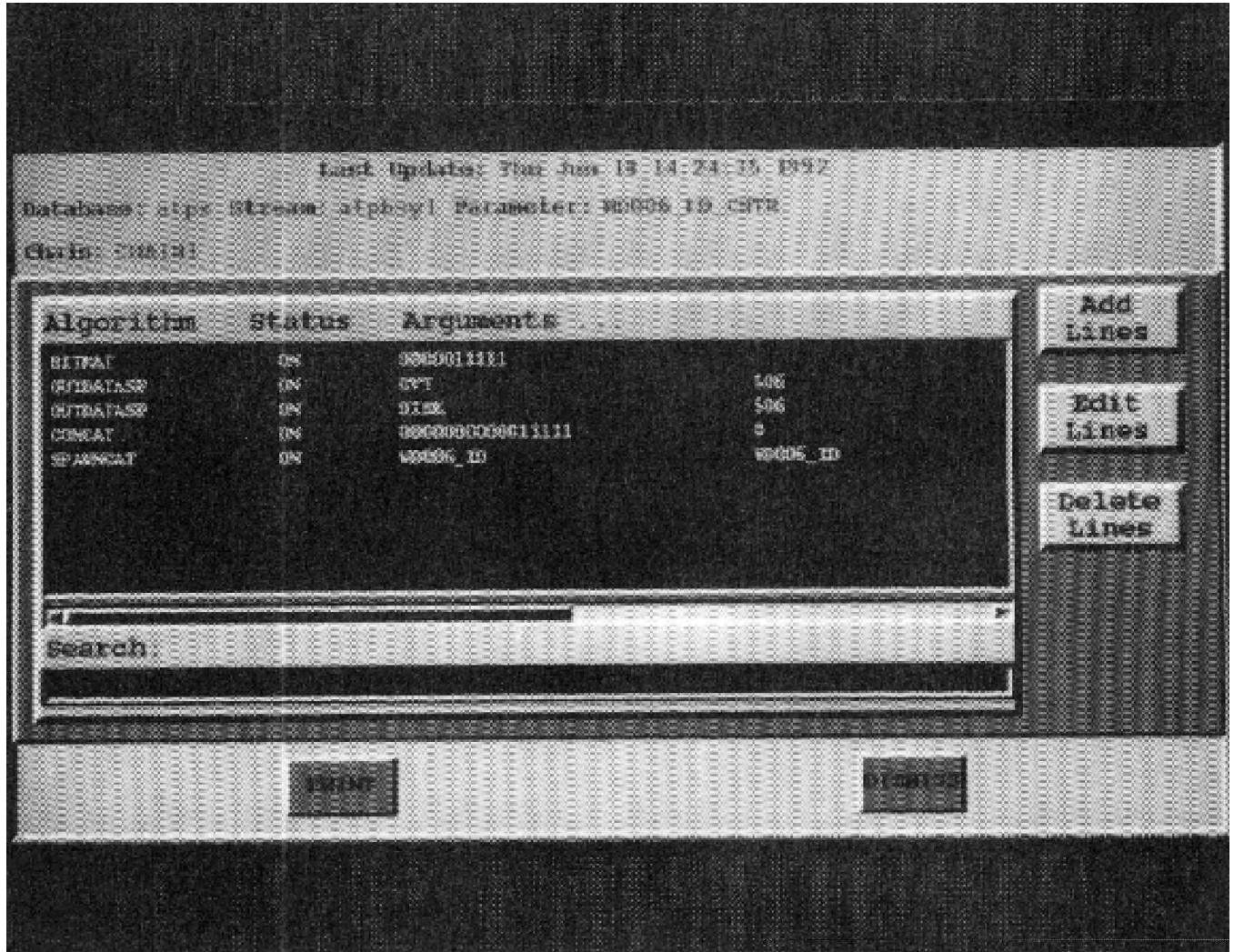


Figure 5. Algorithm Assignments Menu



Specific calculations, such as aileron push rod calculation, may be defined by using a “point and click” method to select derived parameter algorithms from a menu bar algorithm list. Once selected, algorithm arguments are defined through tab selection and keyboard entry. Algorithm arguments may include other parameter names, previously defined constants, bit masks, output tag values and telemetry preprocessor output port selections. These user selected algorithms and argument lists are assigned an algorithm chain name. Refer to Figure 6 for an example of the Algorithm Assignments Menu, which demonstrates special calculation of a left aileron push rod using Telemetry Preprocessor derived parameter algorithms and the following formula.

$$LALPRC = .6667 (LALPRT - 55) + LALPRD$$

Where LALPRT is the left Aileron Push Rod Temperature.

Where LALPRD is the left Aileron Push Rod Pressure in pounds.

Where LALPRC is the resultant Left Aileron Push Rod value in pounds.

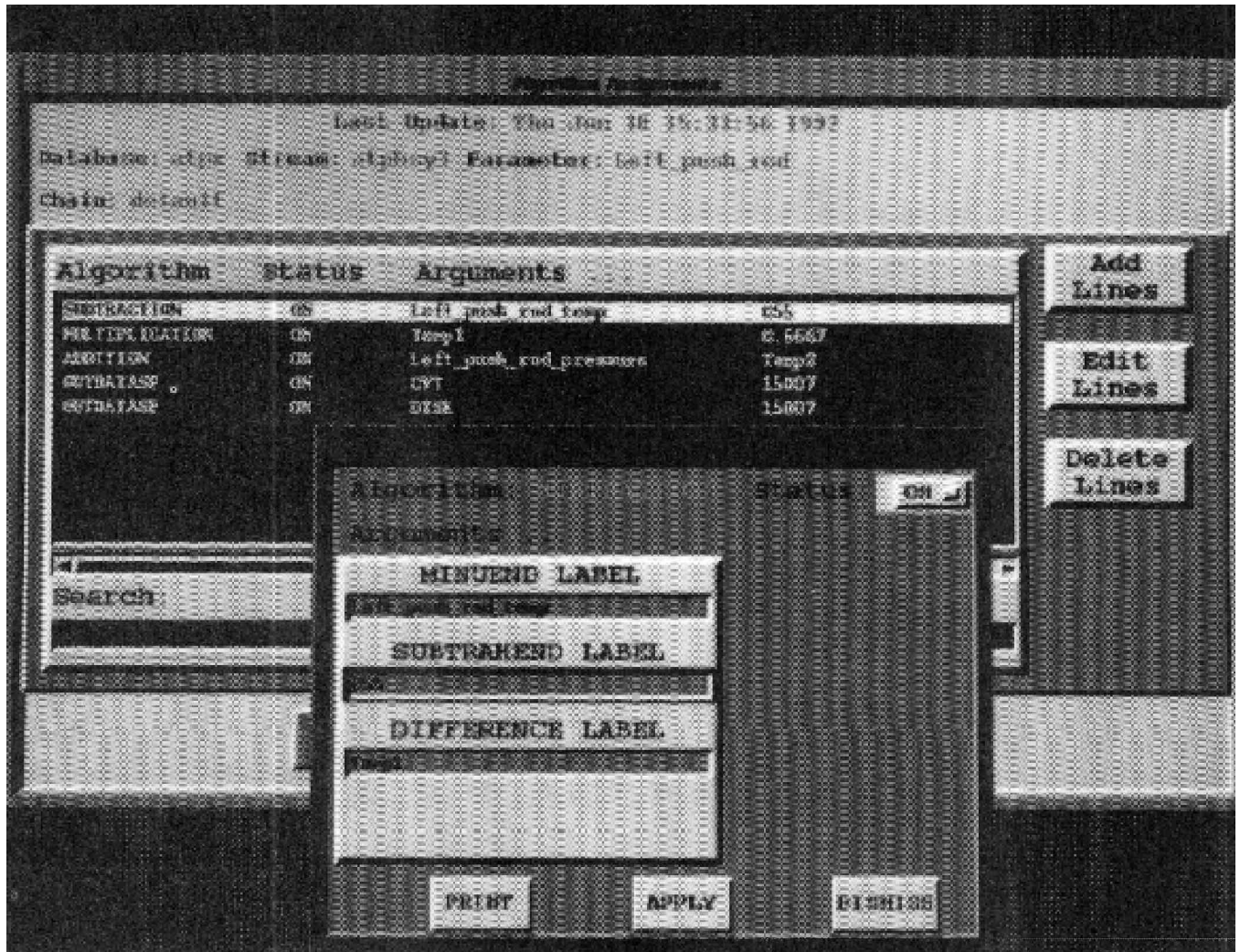


Figure 6. Algorithm Assignments Menu

When database setup is completed, the user selects format compile and download. The assigned algorithms and arguments are automatically downloaded to the Telemetry Preprocessor data processing unit(s). User assigned calculations are processed in the data processing unit(s), and the newly derived parameter values are output through one of the preprocessor output modules. These derived parameter values are immediately available on the user's terminal for display and for output to other devices, such as strip chart via digital to analog converters. The derived parameter values are also stored on the archive device, eliminating the need for recalculation during playback.

## CONCLUSION

The EMR O/S90 approach to creation and setup of real time derived parameter calculations provides unique advantages over other currently utilized methods. A highest order menu level interface that integrates equations, constants, and variables into a working data processing system has been described. Once a specific equation has been defined, it may be reused with or without modifications to create other derived parameters.