N87-11724

THE ROLE OF OPTIMIZATION IN STRUCTURAL MODEL REFINEMENT

L. L. Lehman

Integrated Systems, Inc.
Palo Alto, California

TRADITIONAL STRUCTURAL ANALYSIS/REFINEMENT CYCLE

To evaluate the role that optimization can play in structural model refinement, it is necessary to examine the existing environment for the structural design/structural modification process. The traditional approach to design, analysis, and modification is illustrated in Figure 1. Typically, a cyclical path is followed in evaluating and refining a structural system, with parallel paths existing between the real system and the analytical model of the system. The major failing of the existing approach is the rather weak link of communication between the cycle for the real system and the cycle for the analytical model. Only at the expense of much human effort can data sharing and comparative evaluation be enhanced for the two parallel cycles. Much of the difficulty can be traced to the lack of a user-friendly, rapidly reconfigurable engineering software environment for facilitating data and information exchange. Until this type of software environment becomes readily available to the majority of the engineering community, the role of optimization will not be able to reach its full potential and engineering productivity will continue to suffer.

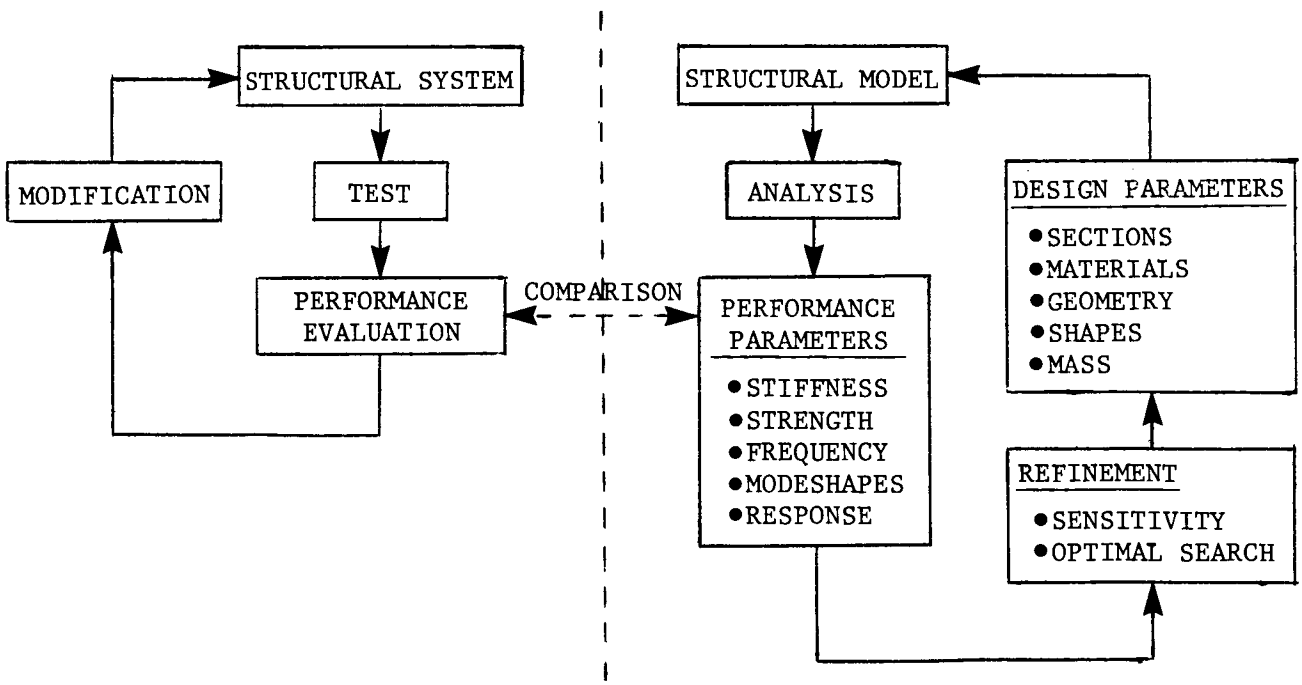


Figure 1.

INTEGRATED ANALYSIS/REFINEMENT SYSTEM

A key issue in current engineering design, analysis, and test is the definition and development of an integrated engineering software support capability. The data and solution flow for this type of integrated engineering analysis/refinement system is shown in Figure 2. Such a system should be capable of providing a wide variety of reconfigurable software tools that support the analysis/refinement cycle and allow flexible data handling. Through careful specification of modular, plug-compatible software interfaces, an engineering analysis system can flexibly support a wide variety of capabilities while providing individual users with a powerful, consistent support environment for their own privately developed software. Support tools that allow user reconfiguration of the application interface should be provided with the core of such a system. Within this type of integrated analysis environment, optimization utilities can be readily combined with analysis tools to more effectively support various stages of the design cycle.

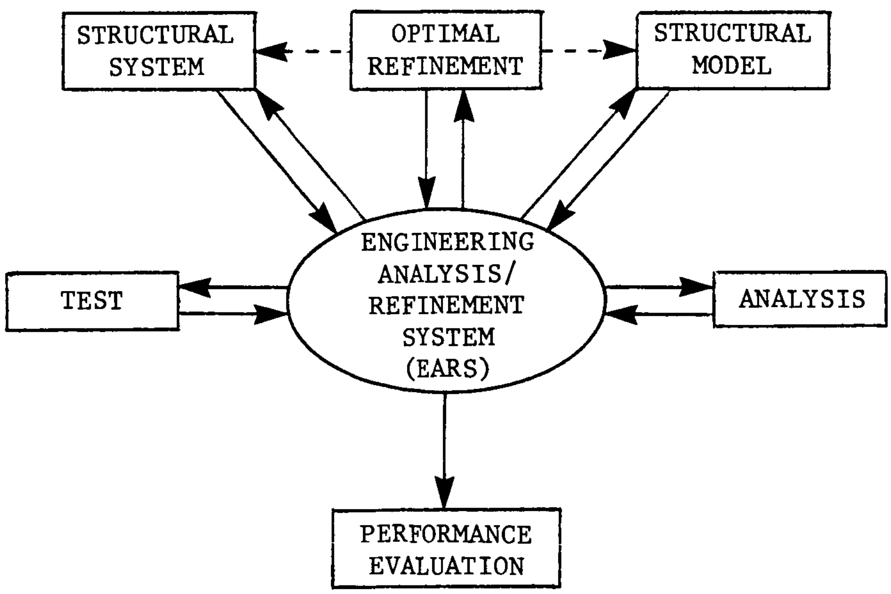


Figure 2.

ARCHITECTURE OF AN ENGINEERING ANALYSIS/REFINEMENT SYSTEM

The general architecture of an engineering analysis/refinement system is shown in Figure 3. The system consists of three major classes of "toolkits" that can interact through well defined interfaces. The toolkits accept input data from the local Engineering Data System and in turn generate output data that is sent back to the local data system. The local data system consists of an engineering data base for permanent data as well as "data stack" handlers that accommodate temporary data. The entire analysis system is driven by a simple but powerful interpretive programming language that controls both execution and data handling.

Within this framework, optimization capability is considered as a general purpose tool that can be used to support the analysis/refinement cycle. An isolated optimization capability is often of limited utility in obtaining rapid problem solutions, but becomes quite useful when it can be readily combined with powerful modeling and analysis tools. In the current context, structural model refinement capability is assembled from a collection of fundamental tools. This type of capability can be made available as part of a preprogrammed (but reconfigurable) application library.

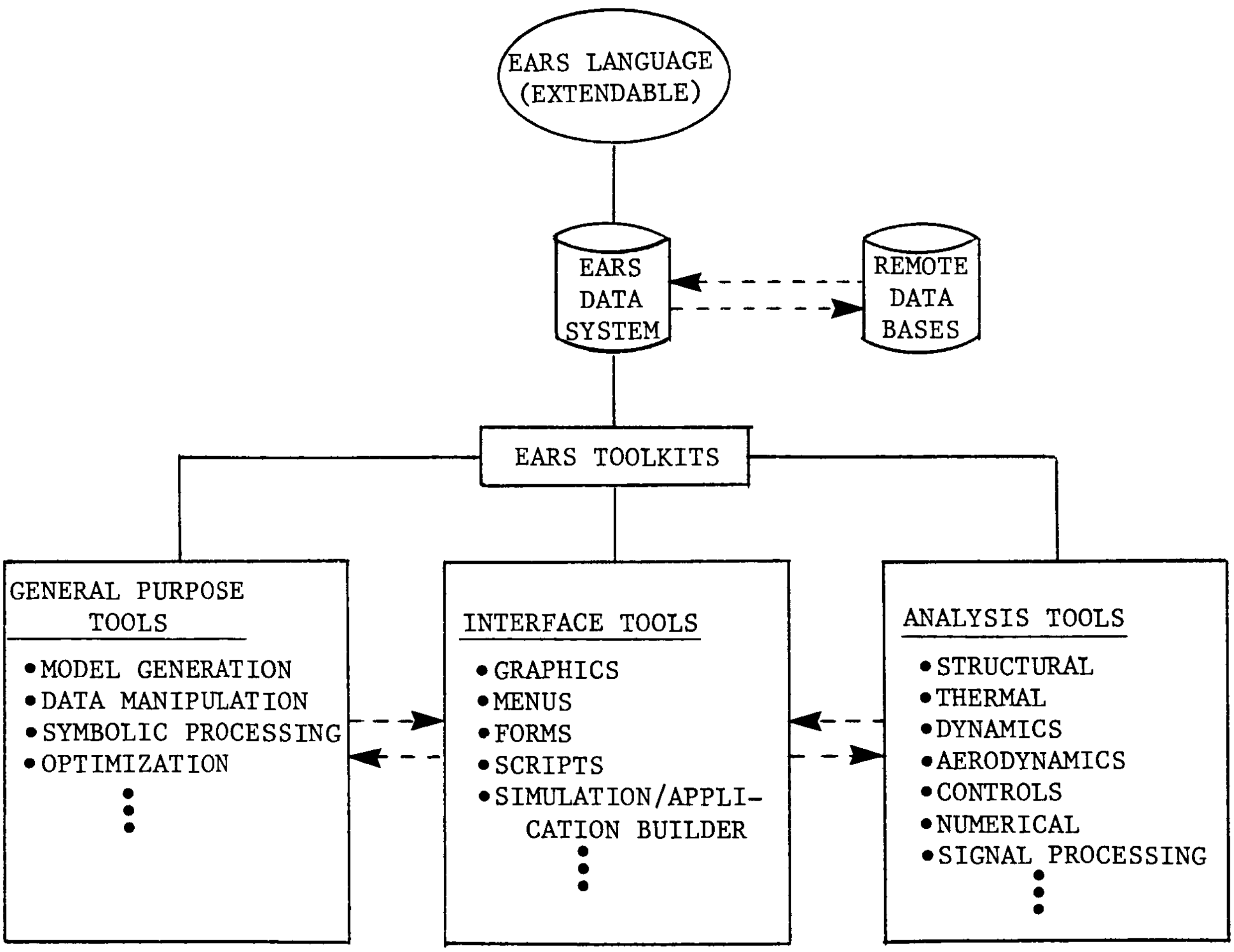


Figure 3.

TOOLKIT SUPPORT FOR STRUCTURAL MODEL REFINEMENT

Support for structural model refinement requires consideration of optimization, structural modeling, and data handling capability. A number of desired capabilities that contribute to the refinement steps of the design cycle are given in Figure 4. As shown in this figure, support for the refinement process is provided both by special tools and by additional capability during the model generation phase. For example, the model generation tools provide not only for various types of models, but also for optimal model reduction and parameter sensitivity. Special capabilities are needed to define measures for model observability and model error. Nonlinear signal processing is needed to detect and quantify nonlinearities in test data before the model identification phase. In addition, special simulation capabilities are required for nonlinear model evaluation.

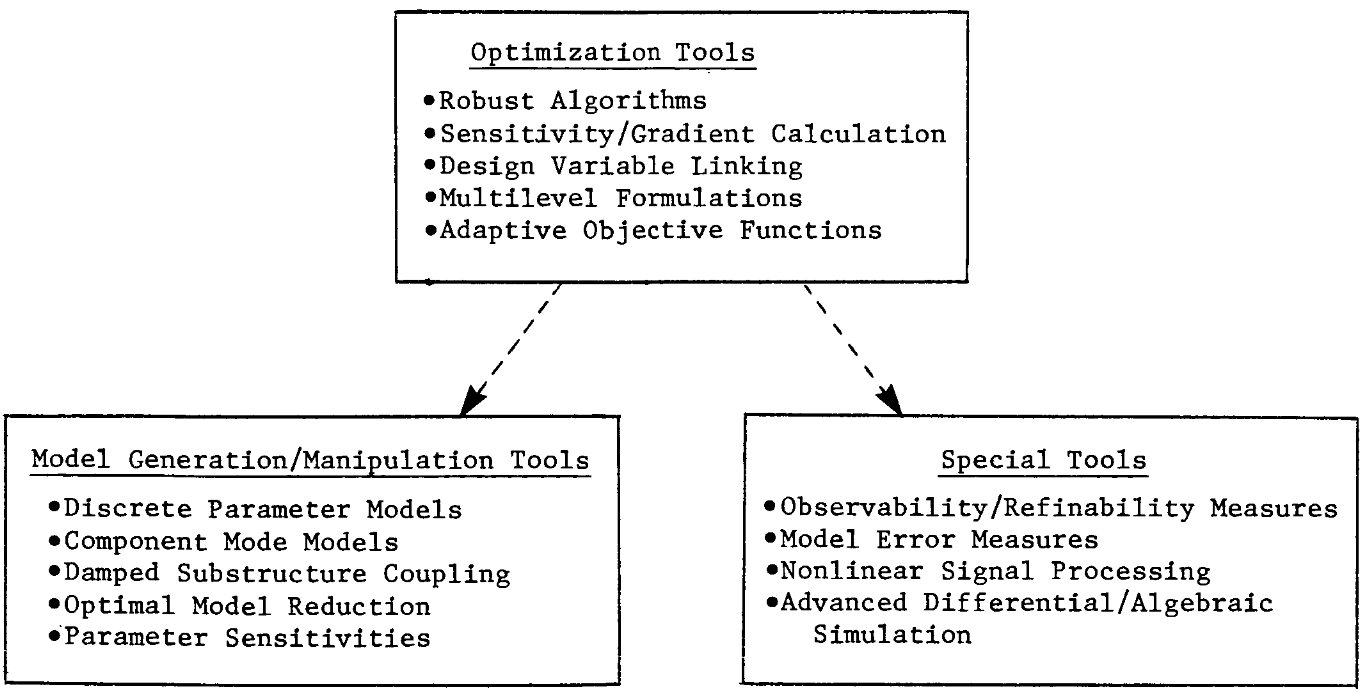
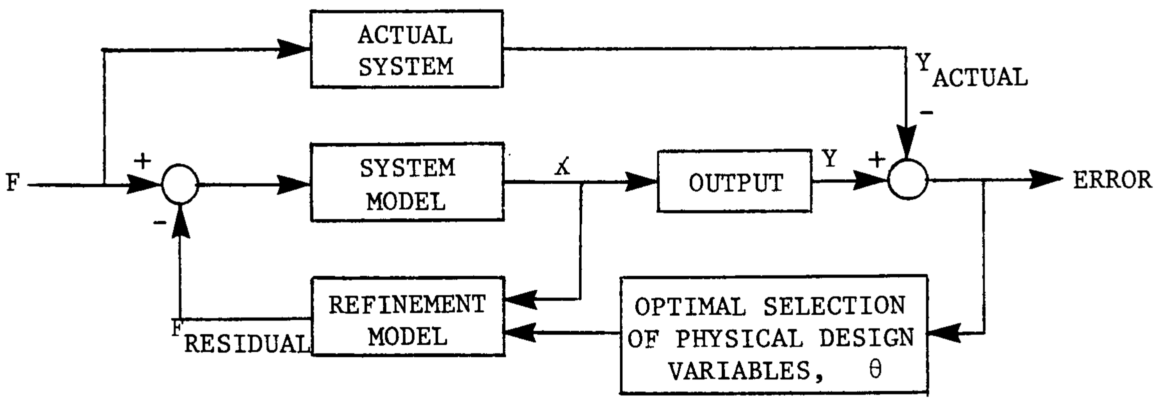


Figure 4.

A GENERAL MODEL FOR STRUCTURAL REFINEMENT

In order to integrate all of the tools needed to support structural model refinement, it is necessary to adhere to a consistent model for both formulation and computation. The diagram of a general model for supporting structural refinement is shown in Figure 5. The main feature of this approach is that parameter refinement is defined in terms of a separate refinement model that isolates the parameters to be modified from the nominal structural model. The refinement model is a function of the physical design variables and the physical system coordinates. Update of the refinement model is achieved by optimal selection of the physical design variables. Output of the refinement model can be interpreted as residuals to be applied to the nominal system model. Definition of the system model is in terms of either physical, modal, or combined physical/modal coordinates. The system model is most often represented in state vector form, which is especially useful for damped systems.



SYSTEM MODEL

$$M\ddot{X} + C\dot{X} + KX = F - F_{RESIDUAL}$$

$$\ddot{Z} + 2\xi\omega\dot{Z} + \omega^2 Z = \mathscr{F} - \mathscr{F}_{RESIDUAL}, \quad X = \Psi Z, \quad \mathscr{F} = \Psi^T F$$

REFINEMENT MODEL

$$F_{RESIDUAL}(X, \theta)$$

Figure 5.

A GRAPHICAL MODEL BUILDER INTERFACE

Much of the effort of combining optimization and analysis capability is associated with constructing, manipulating, and interconnecting various forms of software models. Traditionally, this process has been very time consuming and quite error prone. A graphical model builder based on functional diagrams can provide a far better user interface for model construction and linking. Such an approach can be used for building both analysis and simulation models and rapidly interconnecting them with optimization algorithms to form design and refinement packages. Functional "superblocks" representing either analysis or optimization algorithms can be constructed by assembling desired functions from a catalog of predefined primary functions. By allowing superblocks to be hierarchical, complex capabilities can be built up by following either a "top-down" or "bottom-up" approach. Both user-constructed and predefined functional blocks can be used. To aid in design and refinement of models, functional blocks can automatically provide parameter sensitivities. A schematic depicting the typical construction of performance index and constraint calculation superblocks is shown in Figure 6. By combining this type of problem-oriented graphical interface with flexible analysis "toolkits" containing state-of-the-art optimization tools, significant productivity advances can be achieved in the analysis/refinement cycle.

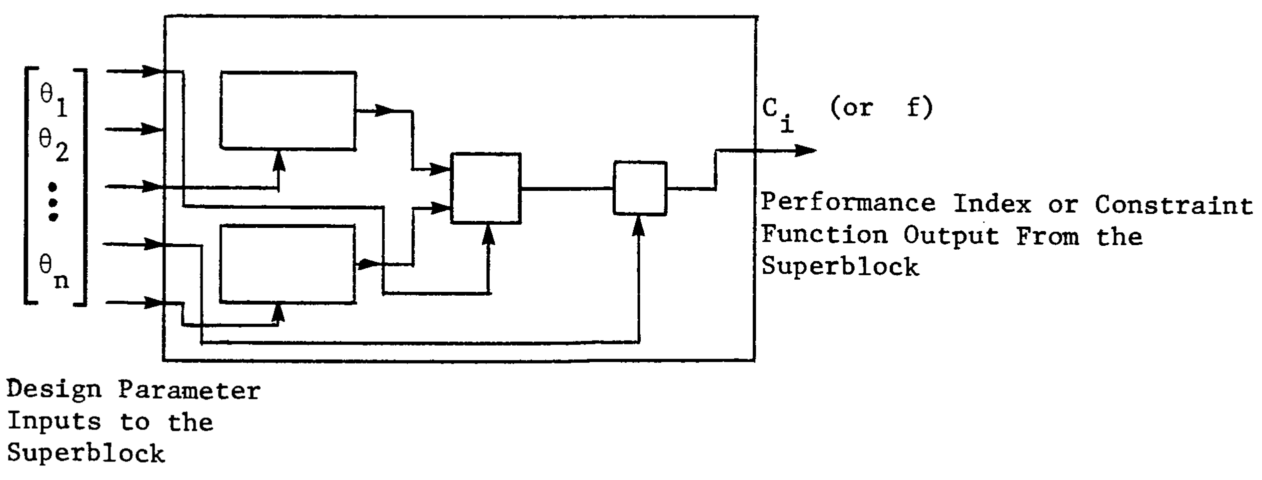


Figure 6.