AD-A255 377

NSWCDD/TR-92/141

# COMBAT SYSTEMS VISION 2030 CONCEPTUAL DESIGN OF CONTROL STRUCTURES FOR COMBAT SYSTEMS

BY BERNARD G. DUREN JAMES R. POLLARD
COMBAT SYSTEMS DEPARTMENT



FEBRUARY 1992

Approved for public release; distribution is unlimited.

NAVAL SURFACE WARFARE CENTER
DAHLGREN DIVISION
Dahlgren, Virginia 22448-5000

NSWCDD/TR-92/141

# COMBAT SYSTEMS VISION 2030 CONCEPTUAL DESIGN OF CONTROL STRUCTURES FOR COMBAT SYSTEMS


BY BERNARD G. DUREN JAMES R. POLLARD
COMBAT SYSTEMS DEPARTMENT




FEBRUARY 1992

Approved for public release; distribution is unlimited.


NAVAL SURFACE WARFARE CENTER
DAHLGREN DIVISION
Dahlgren, Virginia 22448-5000

## FOREWORD

The Naval Surface Warfare Center Dahlgren Division (NSWCDD) is formulating a vision of surface warfare and combat systems for the 2030 timeframe. This effort is intended to provide a framework for combat system engineering and technology investment. A key component of the vision is a functional architecture for future combat systems. This architecture reflects emerging trends in surface warfare, combat system design, and technology. It involves a horizontal organization of weapon systems, plus vertical layers for individual ship and multiship coordination.

This report addresses a control-oriented methodology for combat system architecture work. Related reports include NAVSWC TR 91-607 and NAVSWC TR 91-795, which describe the vision architecture and identify principles of architecture design, respectively.

Approved by:

CARLTON W. DUKE, JR., Head
Combat Systems Department

## ABSTRACT

The centrality of control structures in combat system engineering means that architectures can be derived via existing methods for control of large-scale and complex systems. This report indicates how methods used by system engineers in manufacturing, transport, chemical processing, steel, and distribution systems (for electrical power, water, telecommunications, and resources) can be applied to the problems of combat system engineering. Establishing a theoretical foundation of this type is important because it allows more effective communication and exchange of ideas with experts in both defense and industrial control applications. The underlying purpose is not to invent new theories but to mobilize available scientific and empirical knowledge for application to Navy problems.

# CONTENTS

## CONTENTS (CONTINUED)



## ILLUSTRATIONS

## ILLUSTRATIONS (CONTINUED)



## TABLES

## NOMENCLATURE

| | |
|---|---|
| AAW | Antiair Warfare |
| AAWC | Antiair Warfare Coordinator |
| A/D | Analog to Digital |
| ADS | AEGIS Display System |
| AIC | Air Intercept Controller |
| ASW | Antisubmarine Warfare |
| ASuW | Antisurface Warfare |
| C3 | Command, Control, and Communication |
| C3I | Command, Control, Communication, and Intelligence |
| CAP | Combat Air Patrol |
| CBP | Contractor Bid Package |
| CIC | Combat Information Center |
| CIWS | Close-In Weapons System |
| CO | Commanding Officer |
| CSC | Combat System Coordinator |
| CSS | Communication Support System |
| CUDIXS | Common User Digital Information Exchange System |
| D/A | Digital to Analog |
| DLI | Deck Launched Interceptor |
| DOP | Development Options Plan |
| ECM | Electronic Countermeasures |
| ECCM | Electronic Counter-Countermeasures |
| $F^2D^2$ | Functional Flow Diagrams and Descriptions |
| Hiper-D | High Performance Distributed Processing |
| HOC | Hierarchical Overlapping Coordination |
| I/O | Input/Output |
| IR | Infrared |
| ISAs | Instruction Set Architectures |
| ISS | International Schiffs-Studien |
| LAN | Local Area Network |
| LCF | Link Connectivity Factor |
| LDMX | Local Digital Message Exchange |
| LT | Link Tree |
| NATO | North Atlantic Treaty Organization |
| NAVCOMPARS | Naval Communications Processing |
| NAVMACS | Naval Modular Automated Communication System |
| NAVSEA | Naval Sea Systems Command |
| NCF | Node Connectivity Factor |
| ND | Node Decomposition |
| NFR-90 | NATO Frigate Replacement Program |
| NSWCDD | Naval Surface Warfare Center Dahlgren Division |
| NTCS-A | Naval Tactical Command System-Afloat |
| NTDS | Naval Tactical Data System |
| OPNAV | Chief of Naval Operations |

## NOMENCLATURE (CONTINUED)

| | |
|---|---|
| OR | Operational Requirement |
| OSI | Open Systems Interconnection |
| PDR | Preliminary Design Report |
| PI | Proportional Integral |
| RAM | Rolling Airframe Missile |
| RDT&E | Research, Development, Test, and Evaluation |
| SIC | Ship Information Coordination |
| SRC | Ship Readiness Coordinator |
| STK/ASuW | Strike/Antisurface Warfare |
| STW | Strike Warfare |
| TAO | Tactical Action Officer |
| TOR | Tentative Operational Requirement |
| TLR | Top Level Requirements |
| UWS | Underwater Weapon System |

# 1.0 COMBAT SYSTEM DESIGN

This report considers problems and opportunities associated with a conceptual architecture for combat systems. The architecture itself is the subject of Reference 1. Since the combat system essentially provides a mechanism for control of onboard or cooperating offboard warfighting resources, the characteristic problems of combat system design are essentially those of control synthesis. The conceptual architecture of Reference 1 provides a functional view of this control structure, primarily reflecting end-use considerations. The report is intended to support continued progress toward systematic design of combat systems by:

1. Articulating a theoretical foundation for the conceptual architecture
2. Identifying methods for use in building the simplified conceptual architecture into a more comprehensive framework for preliminary design
3. Identifying concepts and technologies for advanced control structures

The term *design* means different things to different people. As used in this report, the terms *designer* and *engineer* are synonymous. The task of producing engineering drawings is regarded as engineering support.

## 1.1 COMBAT SYSTEM ENGINEERING VISION

For most large enterprises, venture formation is guided by a vision of the future for the domain of interest. The top leadership, with a shared vision and wide knowledge of the domain, decides the timing and direction of investment. Once these decisions are made, the enterprise seeks to build a lasting competitive advantage by delivering goods of such quality that the user is actually delighted. This means products that not only work but are also highly reliable and easy to use, and it also demands a user-centered design process. Past practices, which often were focused more on available technology than user needs, are no longer adequate. Many believe that we are entering an era of fundamental change in productive systems and methods with basic premises as follows:

- Practical limits exist to the economies of scale achieved by large enterprises, regardless of their origin (government or business).
- Information is just as important as capital, land, labor, and material inputs to any productive enterprise.
- The ability to achieve focus (mass, concentration, and coordinated action) in the temporal domain is now as vital as the ability to achieve focus in the spatial domain–in military operations as well as other enterprises.

In a similar way, the Navy must be guided by a vision of the future in its development of new warfighting forces and systems. Existing combat systems, for example, almost universally have been defined so serve a battle organization known in advance. Today there is a growing sense that a certain amount of flexibility should be provided so that operators are not faced with unnecessary constraints at some future time. Other key aspects of such a vision are indicated in Figure 1 below.

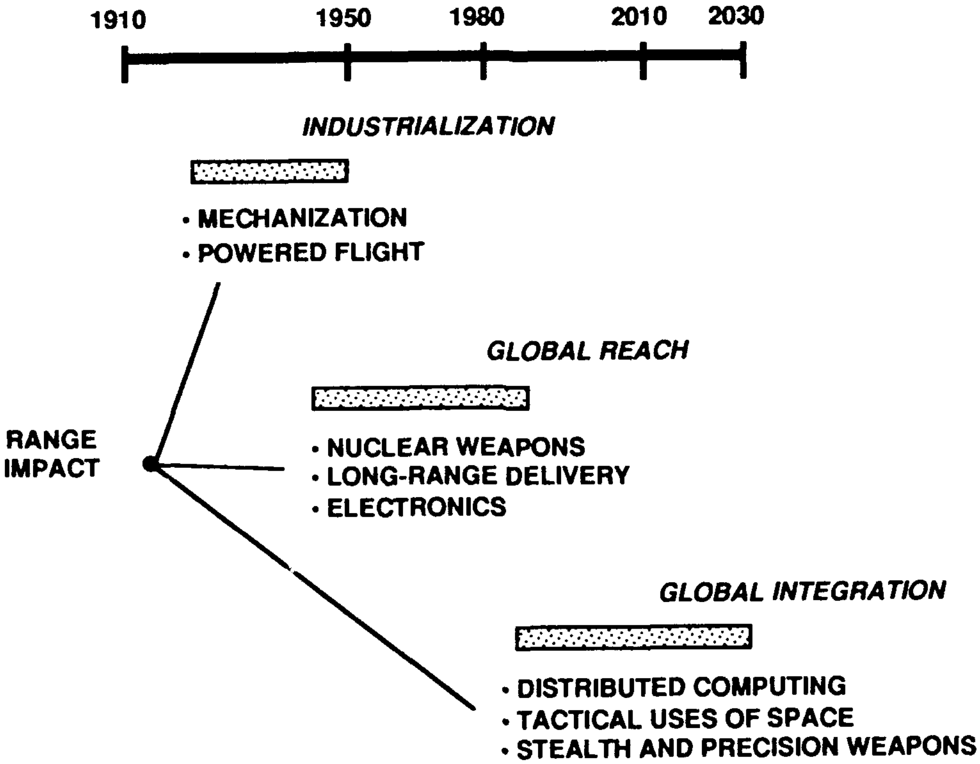



FIGURE 1. LONG CYCLES IN MILITARY DEVELOPMENT

This prospect is a key factor in Naval Surface Warfare Center Dahlgren Division (NSWCDD) efforts to articulate a vision of future combat systems (circa 2030). Concepts for advanced combat systems in particular must be shaped by consideration of future warfare trends as much as technology. The underlving concern is the capacity of the United States to build sustainable warfighting advantages from its raw military and industrial strengths. The Center's role is to assist the Navy to gain a warfighting advantage against possible enemies. This means the

Navy must be armed and equipped with affordable, usable, and effective combat systems that are sufficient to execute the chosen concept of operations against a capable and determined adversary. Just as the best battle plans are conceived in the mind of the commander responsible for their execution, the best combat systems will result from a user-centered design process.

## 1.2 CONCEPTUAL DESIGN PROCESS

Any combat system will go through a birth-to-death cycle referred to as its lifecycle. Six major events in this lifecycle include the following: (1) requirements definition, (2) system design, (3) design implementation or construction, (4) system integration and test, (5) system operations and inservice support, and (6) system retirement. The cycle is illustrated in Figure 2. Reference 2 defines system engineering as the process of translating operational requirements into functional requirements and, subsequently, expanding these functional requirements into detailed equipment and service end item design specifications. Although system engineering spans the entire lifecycle, the focus here is on system design. Several levels of design, or stages in translating requirements into design specifications, are identified, including conceptual design, preliminary design, and detailed system and equipment design.

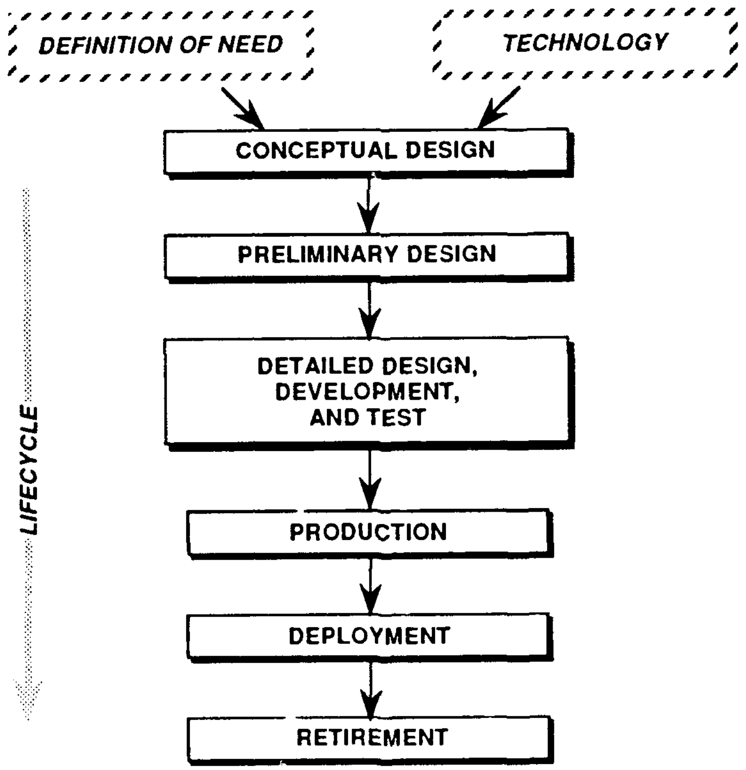



FIGURE 2. SYSTEM LIFECYCLE

The design stages correspond to steps of the acquisition cycle. In the past, *conceptual design* has corresponded to the Development Options Plan (DOP) process, which develops feasible design options in response to a Tentative Operational Requirement (TOR). Figure 3 lists the major tasks associated with this stage of design.

- **NEEDS AND REQUIREMENTS ANALYSIS**
  - *OPERATIONAL CONCEPT*
  - *SYSTEM OPERATING CHARACTERISTICS*
  - *DESIGN CRITERIA, STANDARDS, ETC.*
  - *MAINTENANCE SUPPORT CONCEPT*
- **PRELIMINARY SYSTEM ANALYSIS**
  - *DEFINITION OF MAJOR FUNCTIONS AND FUNCTION RELATIONSHIPS*
  - *INVESTIGATE DIFFERENT TECHNICAL MEANS TO MEET REQUIREMENTS*
  - *IDENTIFY FEASIBLE ALTERNATIVE APPROACHES*
  - *CONDUCT PRELIMINARY ANALYSIS TO DETERMINE MOST APPROPRIATE APPROACH*

FIGURE 3. CONCEPTUAL DESIGN TASKS

The preliminary design stage in turn corresponds to the NAVSEA process for developing a Preliminary Design Report (PDR); see Figure 4. The PDR is in response to an Operational Requirement (OR), which is a specific requirement based on the conceptual design (option) selected by OPNAV. It contains an A-level specification for system implementation that is much more detailed than that of the DOP.

- **FUNCTIONAL ANALYSIS**
  - *DECOMPOSE THESE MAJOR FUNCTIONS INTO LOWER LEVEL FUNCTIONS AND INTERFACES*
  - *DECOMPOSE TO A LEVEL WHERE AN INITIAL (EQUIPMENT) PACKAGING CONCEPTS ARE VISUALIZED (DESIRABLE TO PACKAGE BY FUNCTION) **
- **REQUIREMENTS ALLOCATION**
  - *GIVEN A BROAD FUNCTIONAL PACKAGING SCHEME, ALLOCATE THE TOP-LEVEL PERFORMANCE PARAMETERS, OPERATIONAL EFFECTIVENESS FACTORS, PHYSICAL CONSTRAINTS AND LIFECYCLE COSTS TO THE UNIT LEVEL*
- **TRADEOFF AND OPTIMIZATION**
  - *IDENTIFY UNIT ALTERNATIVES; E.G., NEW DESIGNS OR OFF-THE-SHELF*
  - *CONDUCT TRADE OFF ANALYSES AND SELECT PREFERRED ALTERNATIVE **
- **SYSTEM SYNTHESIS AND DEFINITION**
  - *CONDUCT ANALYSIS OR PHYSICAL TESTS OF THE SYSTEM AS A WHOLE **
  - *COMBINE AND STRUCTURE SYSTEM UNITS TO FORM A FUNCTIONAL ENTITY*

*** INDICATES USE OF STATE SPACE MODELING**

FIGURE 4. PRELIMINARY DESIGN TASKS

Finally, the detailed system and equipment design levels correspond to the NAVSEA process for developing a Contractor Bid Package (CBP). The CBP again is in response to a more detailed set of requirements spelled out by OPNAV in a Top Level Requirements (TLR) document. This represents a specification for system realization, including the actual technologies to be used and how system elements will be packaged and interconnected.

The architecture of a combat system is first developed in the conceptual design stage. Section 2.0 outlines a general approach to conceptual design of control structures. Sections 3.0 to 8.0 address methods for use in the key steps of *conceptual design:* (1) definition of need, (2) functional analysis, (3) requirements allocation, (4) preliminary systems analysis, (5) system concept definition or advance planning, and (6) feasibility studies.

The system lifecycle and the system engineering process begin with the identification of some need arising from a deficiency (real or perceived) in existing combat systems. An ability to clearly state the needs to be met and relate features of the system design to them is essential to communication at all stages of the system lifecycle and in all aspects of system engineering. Procedures that support the process of defining user needs are given in Section 3.0.

The aim of the functional analysis step is to identify operating concepts and functional performance characteristics that can meet user needs for the system. Requirements are decomposed into operational and system functions, and their relationships are identified. These are often displayed in functional flow diagrams. Once the functions have been identified, they are grouped and arranged to form a preliminary system functional architecture. This key result is accomplished by functionally segmenting or decomposing the combat system into near-independent subsystems with distinct functions and well-defined boundaries or interfaces with other subsystems. Operational and engineering principles or rules are also used to guide the segmentation process. Functional analysis may be conducted with the methods described in Section 4.0.

The third step allocates the different performance, integration, and affordability requirements among subsystems of the functional architecture. At this point, the functional architecture not only specifies the functions or tasks to be carried out by the combat system to meet the requirements, but also the manner in which the task is to be performed and how well it is to be performed. Then physical elements (equipment, people, or computer programs) are identified to perform the allocated functions and requirements. At this point, the (initial) implementation architecture of the combat system is developed. Section 5.0 considers methods for use in this step.

The fourth step in developing a conceptual design is that of preliminary systems analysis. Alternative design solutions are identified and screened to eliminate those that are clearly unattractive, leaving only the most promising for evaluation. The problem statement and major risk factors drive selection of criteria for the evaluation process, but it is important to address system support as well as performance aspects. The work is performed in an iterative process and must cover an array of tradeoffs adequate to support the major design decisions that are to be made. Section 6.0 considers methodology for use at this design stage.

The fifth step includes final synthesis and definition of a design solution. Here there is a combining and structuring of alternative design concepts to obtain a preferred system physical architecture. Analysis is normally conducted to ensure that the synthesized design forms a proper functional entity and meets basic performance, integration, and affordability requirements. Essential characteristics of the chosen design solution are thus determined. The final result of this stage is a conceptual design specification. As mentioned above, this leads to further refinement and definition of the operational requirements, preliminary design, and finally, detailed system and equipment design stages. Concepts for use at this stage of the design process are considered in Section 7.0.

Technology opportunities are sometimes considered in an extension of the conceptual design phase. For example, alternative technical approaches for workstations, computers, and communications functions may be considered. Section 8.0 corresponds to this step in conceptual design.

Section 9.0 restates and summarizes key findings and ideas covered in the report. Finally, Appendix A gives engineering principles for time-critical control systems.

## 2.0 SYNTHESIS APPROACH

This section considers procedures for control synthesis based on evolving concepts and practices of control for systems of large scale and complexity. The problem is divided into design of backbone control structures and individual control loops. The major creative challenge in control synthesis is to answer the following question:

> "What are the objectives of control designers in synthesis of control structures, how are they reflected in complex designs, and what is the underlying logic and structure?"

However, this initial question implies a series of other questions that need to be addressed:

1. What is the extent of the control synthesis problem that we want to solve?

2. How are the available measurements to be connected with feasible manipulations to form the various loop configurations?

3. What problem space should be searched for alternative loop configurations?

Reference 3 provides a foundation for addressing these questions by merging methods of process design with control theory. Reference 4 reviews existing approaches to the problem and attempts to identify the potential for future developments.

### 2.1 RELEVANCE OF PROCESS CONTROL TECHNOLOGY

While it may be desirable to address these questions with lessons learned directly from combat system design and operating experience, the existing knowledge base in largely inaccessible. In part, research in control synthesis methods for the flowing process industries (oil, chemicals, power, steel, and telecommunications) is considered in this report as a proxy knowledge base. Problems of combat system design differ significantly from the problems of industrial plant design due to the character of military threats. A key distinction is that the disturbances faced by industrial plants tend to arrive one at a time, while combat systems face an enemy capable of designing complex many-on-one attacks. Thus, it is largely impractical to seek specific control designs for application to both military and industrial systems.

In fact, each of the flowing process industries has its unique aspects. Even within a particular industry, plants are usually unique, with very little replication of specific process configurations and plant control strategies. This means that every application must be analyzed individually. However, this has not prevented the growth of a sizable process control industry and there may be considerable potential for application of basic technologies and engineering principles to combat systems as well.

For the most part, control structures for large and complex systems were developed without any fundamental basis for design save that of hard-won experience. The development trajectory of the flowing process industries (oil, chemicals, power, and steel) is representative. Because process plants are so large and complex, formal and detailed understanding came late and piecemeal. The chemical industry, for example, was mostly a black art until the 20th century, and depended heavily on tradecraft handed down from generation to generation. The modern process control industry began to form only in the 1920s, and Reference 5 asserts that computer process control most likely began with design of analog computers to control the placement of large naval projectiles on a target. These computers, developed by the Ford Instrument Co., contained carefully machined three-dimensional cams for storing trajectory data. Electronic tubes (thyratrons) and later hydraulics amplified gun positioning signals to power levels. Modern control techniques evolved largely from individual, invented solutions to particular problems that were often commissioned by an end user. Systematic experimental studies appeared only in the 1940s.

Many automated flow processes involve time-critical control problems. In telecommunications, switching is a highly time-critical factor. Chemical process operations involve safety concerns that demand fast action in emergencies. In electrical power, safety concerns in the operation of large nuclear power generation plants demand quick response by controllers in emergency situations. All three areas have faced significant technical challenges in the creation of practical and effective control systems.

- From its beginnings, the telephone industry was faced with fundamental problems of great complexity in switching (connecting each phone with every other). Today, network control structures are designed for dynamic management of call routing operations. This involves real-time assessment and response to such disturbances as link breakdowns and traffic overloads. Size and spatial extent of the networks, the demand for many different classes of service, and the random variation of demand and signal factors make this a highly challenging task.

- In nuclear power plants, several reactor vessels and supporting systems are linked to a common control center. Sensors and links supply critical parameters to the control center so that controllers can monitor plant status continuously and take immediate action under time-sensitive emergency conditions. The control structure must provide reliable and error-free performance to prevent severe reactor accidents and to maintain containment of the reactor vessels and supporting systems at all times.

- In many chemical processing plants, various equipment faults, accidents, and natural disasters can produce time-sensitive emergency conditions involving extreme hazards: explosion, fire, and release of toxic chemicals. In refining processes, for example, serious damage can result if an injector is on instead of off or if an unnoticed buffer overflow takes place. Meeting necessary safety standards may involve signal quality monitoring, context checking, redundant coding, or other protective measures. The frequency of major accidents has been reduced over the years, but they do occur.

Thus, the Bell System, General Electric, and Dupont pioneered the use of large industrial laboratories for systematic application of scientific methods to practical problems. The Bell System proved able to apply its skills to development of advanced electronic systems (including elaborate automated weapon systems) during World War II. Military efforts began in 1937 when Bell Labs was asked by the Navy to investigate the possibility of using radar for fire control. A fire-control radar was developed and put into production in 1939. Following this, Bell Labs and Western Electric produced all of the fire-control radar used in Navy ships during the war years. The M9 antiaircraft equipment, one of the greatest advances in fire control made during the war, was also produced. The M9 Director provided a functionally complete system for target destruction capable of acquiring and tracking targets by radar, computing gun positions, and issuing control signals to the gun mounts. All together, the Bell laboratories handled over 2000 separate defense projects, including major radar and gun direction systems, specialized communications equipment, sonar, proximity fuzes, magnetic mines, acoustic torpedoes, and advanced materials research.

## 2.2 PROBLEM SCOPE AND FEATURES

Essential features of the problem of control structure synthesis include the following:

- In conceptual design, consideration is limited to basic operational requirements, such as attaining resource balance and cornerstone performance criteria at various loadings, guaranteeing adequate performance within some range of stress levels, and achieving low interaction levels among the various units.

- The controlled behavior is examined in terms of key design variables or key structural characteristics of the system. The aim is to consider the feasibility of various alternative process designs and to determine the steady-state capacity and structure needed for effective control. Thus, only simple steady-state balances and equilibria need to be considered. General and simple measures of operability and quality of achievable control are in order. The synthesis approach should be simple and quick while allowing latitude for engineering judgment. Extensive computations and detailed dynamic analysis are typically not affordable.

- What is essential at this stage is to identify key controlled variables and how they are affected by design or structural changes in the combat system. Consequently, sensitivity analysis is crucial. Various techniques can be used to evaluate sensitivities.

- Only primary manipulations and measurements should be considered, particularly those related to basic operational variables.

- The loop structures should be simple and operationally feasible. At this stage, complex and detailed configurations are of little interest.

Control synthesis problems are addressed at three different levels: (1) individual operating units or subsystems, (2) plant backbone control structure, and (3) complete plants. The first corresponds to engagement control at weapon system level and is beyond the scope of this report. The others are addressed below.

## 2.3 PLANT BACKBONE DESIGN

Since the 1950s, naval combat systems of the North Atlantic Treaty Organization (NATO) allies, including the U.S. Navy, have evolved from a bottom-up design approach. The approach is roughly as follows:

- Components are developed independently by many program offices. The basic concept is to build a little, test a little and check out, debug, and then produce a very specific element of the system.

- The components are procured as commodities and integrated into a combat system; communication between systems is achieved by dedicated, hard-wired computer input/output channels.

- Creating a valid combat system of components with distinct hardware, software, protocols, and sponsors becomes a major effort.

- Additional capability and functions are added in time as the threat increases. Additional computers and computer input/output channels are provided to integrate more subsystems.

The basic architecture could be considered *federated*. During this period, the cost of computers and memory (in megabits/second) has become much less expensive, but the cost of computer programs has gone up. Federated systems are difficult to change because they are achieved by computer-to-computer interface and system integration requires extensive computer programming effort. Substantial time and testing are also needed to certify that changes have not affected the existing computer programs.

A design strategy called Dynamic Process Control evolved during this period, and with its formulation in Reference 6, the strategy came to be accepted as the dominant approach to control systems. Control actions were divided into two categories: those needed to achieve material balance control throughout the plant and those needed to maintain product quality control for each individual processing unit. The former was necessary for plant management in the presence of low-frequency changes like production scheduling and could be designed separately from the latter, which was intended to regulate high-frequency disturbances entering the various units. The proposed strategy was to design plants with sufficient storage that the break frequencies of the resource balance control systems (low-pass filters) were an order of magnitude lower than the resonance frequencies of product quality control systems (high-pass filters). The two control systems thus worked in sequence, yielding a decomposition permitting individual and separate control designs.

Dynamic process control was not only employed for design of plants from an operability point of view but also constituted the primary basis for synthesis of plant control structures. The inherent decomposition of the material balance and product quality control systems led to a clear identification of operational control objectives. Essentially, large overdesign margins, storage buffers, and controlled operating conditions were used to prevent effects of large disturbances from travelling through the plant. Under this regime, plants experience only smooth operating conditions and regulatory control problems are tractable. Consequently, control systems for single operating units were the principal focus of designers. This gave a bottom-up design approach, but the number of alternative control configurations remained large. Final selection was usually based on experience, on rules of thumb, and on dynamical simulation of the plant to be controlled.

This strategy worked very well, serving the control needs of thousands of plants, until two important economic constraints came into play. First, shortages of raw materials and energy appeared. The shortages led to new integrated designs with better energy management and more use of recycle flows in the plant. Second, restrictions appeared on fixed capital expenditures. The reduced capital expenditures led to lean processing units with very small overdesign margins and the elimination of intermediate storage. Operating units thus came to be tightly coupled, with strong interactions through energy recovery systems and recycle flows. This made it difficult to maintain a process at desired set points while providing also for safety, robustness, and steady economical operation. Therefore, the dynamic process control strategy is not suitable for many of today's plants and their successors.

## 2.4 COMPLETE PLANT DESIGNS

Due to the new levels of integration needed in contemporary plants, process control technologists have given increasing attention to the problem of synthesizing control structures for entire plants as opposed to individual processes. The starting point for this is an approach formulated by References 7 and 8 that draws heavily on Buckley's pioneering work, plus subsequent extensions and refinements. It retains the division of control objectives into resource balance and product quality categories used by Buckley, but does not impose this decomposition on plant control structure. Two steps are involved: the first step develops a plant-wide material and energy balance control system allowing smooth and efficient changes in the production level, and the second step synthesizes local steady state controllers for the individual processing units, which either complement the resource balance control system or satisfy product quality requirements.

This approach has the right characteristics for use in conceptual design and may offer a valuable guide to synthesis of combat system control structures. The resulting approach can be given as follows:

1. Seek opportunities for problem decomposition both in the process and its operational objectives. Order of magnitude arguments may be used to discover the most important class of disturbances, identify corresponding control objectives, and determine the sensitivity of controlled outputs to various measurements and manipulations. The aim is to establish a decomposition strategy that exploits inherent problem characteristics, is compatible with operator training, and makes good use of digital hardware capabilities. Since different sets of control laws are available for different control configurations; however, synthesis of the best configuration of controllers remains a difficult task.

2. Determine the backbone of the final control configuration; i.e., a configuration that will handle plant-wide operations by coordinating single unit operations, leaving control loops for unit operations to a second stage of synthesis. The problem of system decomposition must be considered here as an inherent part of the design strategy.

   For combat systems, the concern is not so much material balance as it is overall load balancing (target inputs, time, energy, throughput, ordnance, and manpower). Steady-state sensitivities among the various load variables should be computed to guide decomposition. The options available for subsystem control include a wide range of standard control loops, but this backbone design problem is largely unsolved.

3. Synthesize the control loops for local unit operating control needs.

   Individual control loop design involves the following special cases:

   - Single units with fixed loop structures
   - Single units with unspecified loop structures
   - Optimizing control structures for single units
   - Variable control structures for single units
   - Secondary measurements/manipulations in single units

   Subsystem or local unit configuration alternatives include feedback, feedforward, cascade, decoupler, ratio, multiple output loops, redundant loops, switched controls, inferential, and other designs.

4. Where critical features of the design are concerned, dynamic simulation studies can be performed in order to verify whether the candidate control structure will be adequate.

During the course of the above synthesis procedure, it is necessary to make a number of simplifying assumptions and to use simple heuristics to select the best of several control alternatives.

Although it may yield a control design that is unable to completely damp out all fluctuations, this approach is appropriate for use in the preliminary design stages where the main concern is feasibility of operations against the major disturbance classes. Subsequent work by

Reference 9 suggests this approach can be extended to include several layers of control. For conceptual design of combat systems, the following layers are envisioned.

- *Balance Layer:* The design process begins with careful consideration of likely mission profiles, threats, and scenarios. Since combat systems are considered in this report as plants for the processing of targets, the balancing of weapons and targets is a primary concern. Cornerstone factors such as reaction time, firepower, coverage, hardness, and sustained readiness are driving factors. Balance across the warfare mission areas is a second area of primary concern. Capabilities and ordnance loadouts should be brought into balance across the warfare mission areas. Layered defense concepts introduce a third key area of concern since they involve balancing of resources across the layer structure. Once these concerns have been addressed and appropriate control objectives formulated, the availability of useful measurements and manipulations is considered. A dominant theme for control design in this layer is to establish loading and balance controls for the main operating tasks and performance features of the combat system. Space and time, bandwidth, radio frequency, material, functional, and energy balances are among the resources to be considered.

- *Path Quality Layer:* Readiness and performance quality controls would be considered as a second layer of control problems. Here the primary action paths for projected single and multiple engagements are probably the main concern. Kinematics, time, and error budgets will be among the problem drivers at this stage of design. As much as possible, actual threat and weapon performance data should be relied upon to determine what manipulations are paired with what measurements. Secondary action paths, including backup and alternate operating modes, are addressed after the more critical controls are defined. Secondary paths need not be fully considered in conceptual design but should have full consideration in the preliminary design phase.

- *Facilities Layer:* A series of functions with multiwarfare significance are viewed as a third layer of control problems. Navigation and tracking, multipurpose launch systems, communications, tactical computing, air operations, combat information center (CIC) configuration, and signature control are among the factors that are driving combat systems to more tightly coupled (integrated) operations. The long-term trend is for combat systems to have a backbone control structure so that operational tasks can be accomplished with minimum reliance on coordination and control assets uniquely identified with individual warfare mission areas.

- *Coordination Layer:* Here, the factor of concern is interaction rather than balance among the alternative paths for processing of targets. The aim is to coordinate target processing for optimum overall performance. This applies to weapon clusters with specialized tasks such as short-range antiair warfare (AAW), to entire warfare mission area systems, and to the combat system as a whole. In addition, certain target complexes may be of interest. For example, threat fast attack craft and submarines acting with a degree of tactical coordination may call for simultaneous and coordinated action across all primary warfare mission areas. The potential for undesirable interactions between elements of the combat system (e.g., electromagnetic interference) should be considered at this stage of design, and in some cases, specific design features may be added to decouple the systems involved. Constraint controls or overrides are generated at this stage of design to maintain operating conditions within the allowable boundaries of operational feasibility.

- *Ship Layer:* This layer concerns load balancing and allocation of volume, deck area, weight, moment, utilities, and interconnections for combat system equipment at the ship level. Startup and/or restart controls are also considered. This stage of design represents a point of coupling between combat system design and overall ship design.

Throughout this layered design process, factors with a major influence on the overall military worth of combat systems must receive careful consideration. Key drivers include the following:

- Technological transparency to users and *evergreen* or extensible system design for ease in adding performance or functionality
- Open system architecture for ease in creating new interfaces
- Self-revealing design for ease of use, production, and support
- Ability to distribute computational resources
- Embedded command support (relative to decision-making cycle time and process quality)
- Connectivity (relative to orders, information, and control)
- Environmental immunity of implementation

Solutions will be guided partly by engineering, cost factors, and control theory considerations. For example, if only single loop controllers are employed, then manipulated variables must be chosen to minimize interaction between the loops. Use of multivariable control methods would allow somewhat more latitude in design. At this stage, it is important to push for process integration and consequently lower venture costs without endangering process operability. Key economic tradeoffs between development costs and operating costs are also considered.

## 2.5 ADVANCED DESIGN STAGE

Once a structure has been selected from among the many alternatives, a control system must be synthesized at a much more advanced stage and with many more operational details. System dynamics must be considered as well as steady-state control, and a more comprehensive treatment of basic control objectives becomes necessary. In particular, a set of control objectives that covers both normal and special purpose operations should be considered. Even normal operations may involve abrupt change in control objectives, including interrupts as well as shifting priorities. Special purpose operations include startup, shutdown, and changeover of operating processes as well as abnormal conditions that place the combat system at risk. Thus the following issues must be addressed:

- Translate all operational requirements into control objectives and identify all corresponding controlled variables.

- Identify all relevant measurements (both primary and secondary) for inferential control or for improved dynamic behavior.

- Identify all manipulations (both primary and secondary) for improved dynamic behavior.

- Determine the best structure of pairings between measurements and manipulations in feedback, feedforward, inferential, cascade, or other loop configurations.

It is clear that such multiobjective problems, which demand quality solutions, are highly difficult and challenging. Here is where novel and imaginative formulations are most needed. At present, the lack of suitable design aids often means that an inspired guess must be made in the initial design stages. Once the detailed design phase has commenced, it is far too time consuming and costly to attempt to reconfigure the system.

Completing solution of the problem in the final design stage involves the following additional issues:

- Safety: selective features, override control loops, etc.
- Optimizing control procedures: searching for a new optimum level of operation, implementation strategies, and variable control structures.
- Startup, restart, shutdown, and changeover control procedures.
- Control procedures for emergency situations.

At this stage, many of the critical questions have been resolved; what is left, however, still demands careful and methodical work.

## 2.6 PERSPECTIVE

These are only the initial steps toward development of the highly integrated operations expected in future productive systems. Since the 1940s, the world has been split between capitalism and communism–east and west. Many now envision a world that is split between fast and slow economies. This vision is premised on the evolution of a single integrated loop connecting all the key parties to a commercial transaction: customer, producer, distributor, and payment agency. Underpinning this loop is a system of lean production that uses less of everything compared with past mass production systems: half the human effort in the plant, half the space for manufacturing, half the investment in tools, and half the engineering hours to develop new products in half the time. Plants would be designed in a top-down fashion to gain efficiency levels bordering on perfection and producing continually declining costs, zero defects, zero inventories, and endless product variety.

As these economic and technological trends mature, a shift to top-down combat system design methods seems highly likely. The course of the NATO Frigate Replacement Program (NFR-90) clearly points to a shift in this direction. All of the nations involved wanted a structured approach to combat system functional design based on the operational requirements of the warfare systems and the threats. Program requirements therefore called for a top-down approach to functional design of the combat system. This was to include explicit specifications for (1) how those functions were to perform, (2) where they would be located within the system, and (3) how they would relate to other functions that must be performed.

## 3.0 IDENTIFYING USER NEEDS

This section addresses the problem of identifying user needs, an aspect of combat system engineering that deserves consideration as a way of improving design quality and strengthening the system engineering process at the same time.

A basic principle of combat system engineering is that design must be user centered: that is, the *design* objectives must be aimed to help users achieve the *operational* objectives for which they are responsible. What makes user-centered design difficult is that it may take five or six attempts to get a design right. Since users will not accept a product that does not work right, necessary testing and refinement cannot be performed after release of the product. The best solution is thus to break out of a straightline development process to allow users to test early descriptions and prototypes of the product system (several times if need be) and then refine its design in response to their experience. The final design must still be supportable but, in general, the initial design should concentrate on the user for whom the system is targeted. Identifying end use environments is also a concern, since the new system must fit in comfortably with systems and processes already in service. The necessary information can be gathered by observing users working at tasks that will be affected by a new system, by conducting interview, or by "walking a mile in the user's shoes" as a participating observer.

The underlying concept is to create a user-centered design process based on early and continuous efforts to identify and validate user needs. Five steps are essential:

- Identify and understand potential users.
- Understand user environments and tasks to define all functionality needed in operations.
- Formulate alternative service concepts and assess how new systems or technology would benefit the users. Cost, performance, data, technology constraints, and capabilities of other user equipment are considered in this step from a total system point of view.
- Analyze current strengths and opportunities as well as key areas in which better quality would improve user satisfaction. How each function should be provided is also determined in this step with prototypes and trials to gain critical user feedback and refine the design concept.

- Apply this knowledge to create or modify systems and procedures for delivery of quality in combat systems.

Combining the lessons learned from all five steps will enable combat system designers to identify the small number of product features that are vitally important. This feature set establishes the key design dimensions.

## 3.1 MISSION AND OPERATING CONCEPT

Planning for naval forces occurs at four primary levels. The first (global) considers the overall character and contribution of naval forces to the national security. The overall mission and role of the service, its design or architecture, the methodology to be used in force planning or design, and the performance standards to be met are key concerns at this level.

The second level (force planning) takes into account the likelihood and severity of potential armed conflicts, forecasts of military technology, and national security policy. At this level, decisions are made about what mission capabilities to pursue and what technology to introduce.

The third level (component forces) provides for implementation of Navy force planning decisions. The chief concerns are the numbers and capabilities of operating units needed, the potential uses of new technologies, and the budgetary choices that must be made. For example, new construction and modernization programs for surface combatants are formulated at this planning level.

The fourth level (program elements) provides for realization of needed changes in system capabilities and configurations. Resource allocation decisions are the chief concerns at this level. For example, a short-range AAW missile may replace an existing AAW gun system in a new combat system baseline configuration.

Combat systems must be designed to support a general concept of operations with minimum dependence on operational or implementation details that cannot be reliably foreseen. Defining a reference model for combat operations makes a reasonable starting point. This model should be framed to capture basic missions and operating concepts rather than to represent specific engagement details. It must provide for definition of key concepts (e.g., entities, systems, and interactions) and a structure accommodating relationships between the defined terms. For completeness, these relationships should encompass all actions that may be expected in any given operating environment. In addition, this reference model should reflect planning concepts and doctrines for the conduct of naval warfare as they relate to the employment of combat systems.

The choice of mission as a key organizing factor for combat systems is an important step. Notions of centralization and decentralization are closely linked to whether an organization is structured in terms of mission or function.

Organizing by mission reduces the opportunity for exploiting economies of scale, and there is a possibility that the mission-oriented subgroups will become parochial in outlook. In highly decentralized command structures, ambiguity and the need for unstructured problem solving by lower level personnel may lead to degraded morale or performance. But mission decomposition substantially reduces coordination requirements, provided the missions can be carried out in an relatively independent manner.

In this approach, the system is viewed as a black box that has well-defined responses to external stimuli. The requirements engineer identifies discrete incidents that occur in the environment of the system and defines the responses to these events as separate processes. This approach has the advantage of forcing the developer into the specifics of the application. Once the basic model is built, it is easy to add response constraints and to construct acceptance tests. A severe disadvantage is that the model is unstable in the face of changing requirements.

Functional decomposition permits economies of scale due to specialization, but there are very few missions that functionally organized units can perform by themselves. Ordinarily, a higher level decision maker must provide overall coordination and control, or if the task is a routine one, standard operating procedures may be sufficient. Organizations that are functionally structured from top to bottom are likely to incur heavy coordination costs, whatever the means used. Thus, such organizations tend to be less able to adapt to rapidly changing situations.

The primary alternative to functional decomposition is to organize by mission at the top and by function nearer the bottom of the hierarchy. Most organizations use such a mixture. The main question for design is then at what level to segment on missions and at what level by specialty and function. Another key question is how the specialty subunits can be brought into the service of the mission-oriented units.

## 3.2 ACTION PATHS

A combat system may be viewed as a plat for the processing of targets. Value is created by execution of warfighting processes under orders. Each warfighting process involves a string of discrete actions for functions designed and sequenced to achieve a significant combat task. Such strings of discrete actions or functions are termed action paths. Since many different action paths are employed, each critical to some aspect of mission performance, combat systems are designed around the diverse action paths that must be produced.

A set of generic action paths is shown in Figure 5 below. The structure of action paths is addressed in Table 1. For a given scenario and set of mission packages, the problem of combat system design can then be approached as follows:

- For each given scenario and mission package, start with a generic action path and proceed to construct functional descriptions tailored to the given scenario and set of mission packages.
- Identify existing or projected functional options for implementation of each action path (whether complete or fragmentary), considering each path element in turn.
- Examine the potential for resource sharing or contention, interference, and support between the action paths.
- Use information gained about path interactions to identify underlying structure within the combat system and to evaluate candidate architectures.

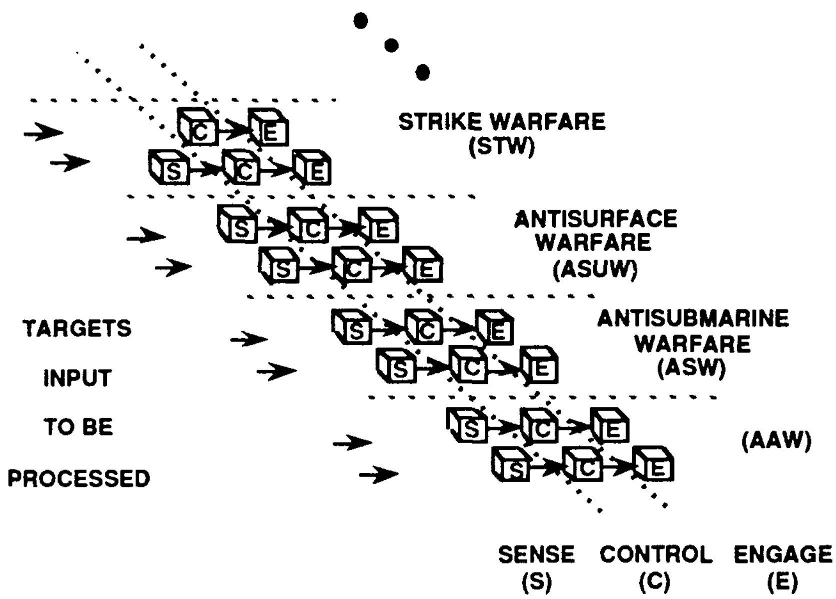



FIGURE 5. TARGET PROCESSING PLANT

TABLE 1. GENERIC ACTION PATHS

| INFORMATION PROCESSING ACTIVITIES | STATES OF KNOWLEDGE |
|---|---|
| 1. Detect need for action | Alert generated |
| 2. Observe information and data | Set of observations |
| 3. Identify present state of system | Deviation from target |
| 4. Assess effects on current task | Closure uncertainty |
| 5. Evaluate performance criteria | Goal set revised |
| 6. Assess effect of goal change on current task | Target state identified |
| 7. Define task; select desired change of system condition | New task specified |
| 8. Formulate procedure; plan sequence of actions | Procedure selected |
| 9. Execute; coordinate actions | Actions completed |

Action paths can be organized by warfare mission area. When this procedure is used for strike warfare (STW), the results are as shown by Figures 6 and 7 below.

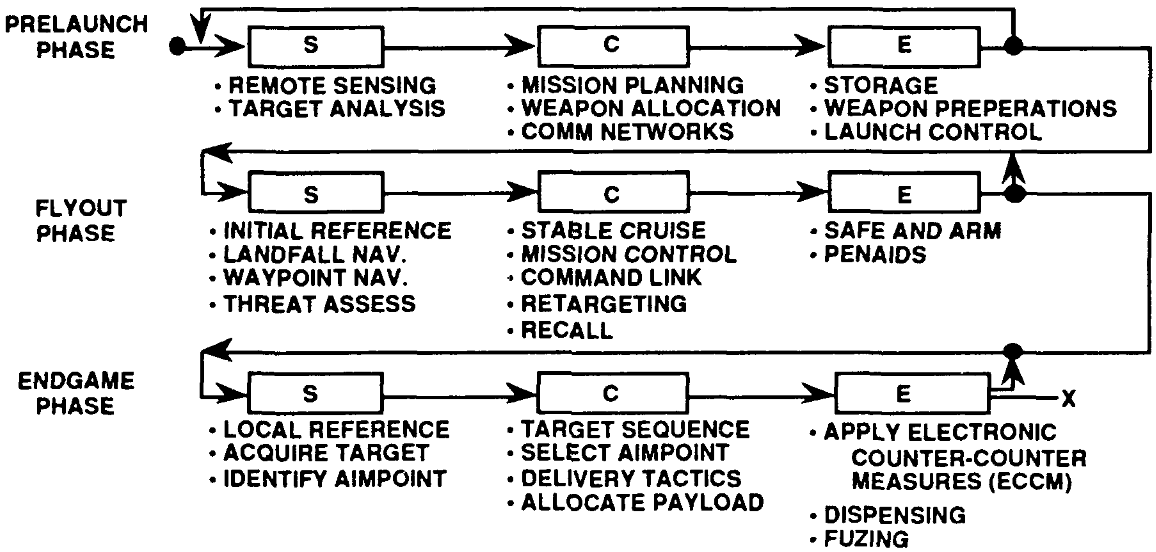



FIGURE 6. STW ACTION PATH

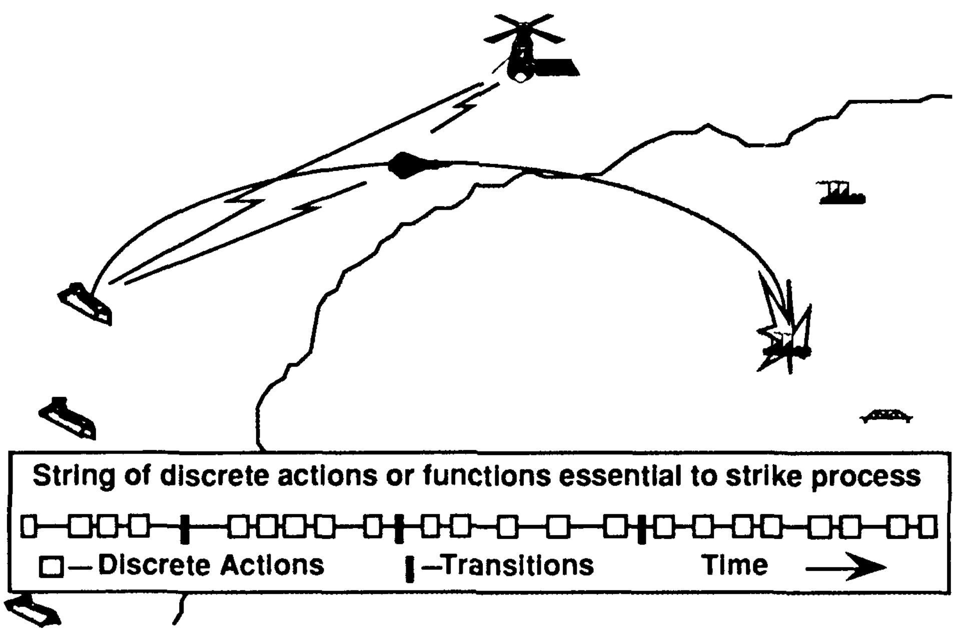



FIGURE 7. STW SCENARIO

Only when the concept has been tested and all relevant groups of users identified can the system be formally defined. In any case, continuous dialogue is necessary because the user is the one who best knows the process to be controlled. Where possible, it is advisable to take a user into the project team. This requires that the requirement specifications can be understood by the user and that the models can be used in the dialogue. The definition must address not only all functions required by each type of user, but also such factors as how users learn to utilize the system, how it is obtained, how it is installed and maintained, and how it is controlled.

A primary advantage of computer aids is the creation of a database that can be accessed by the requirement specification team and that also indicates items not yet defined so that they cannot be forgotten. Sometimes a high-level model of the real-time control system and its environment can be generated, provided the execution times of actions (including the real-time executive) are known. Such a model could be executed to produce simple actions in response to simulated stimuli. More complex outputs can be generated by cascading these simple actions, with each action setting conditions that trigger other actions until a realistic scenario has been assembled. Such execution is often called rapid prototyping and is the first step of what can be characterized as incremental development.

Without clear standards for the process of generating requirements, designers face several problems in identifying and verifying user needs.

- Without a clear statement of methodology to guide preparation, requirements tend to vary in technical content. Some will be overspecified, while others will be vague or even incompatible.
- Unless formal procedures are established for identifying or validating user needs, user input is often lacking or unsatisfactory.
- Once specifications are written and released to designers, there may be no formal mechanism of change control. Indeed, redundant documents may be created because designers have gotten used to doing their own system engineering.

An ongoing quality improvement process is needed to overcome these problems. Process quality metrics can be defined in terms of timeliness of document delivery; amount of user input; percent of documents requiring no rework; reviewer judgments of quality, considering readability, accuracy, clarity, completeness, and treatment of open issues; responsiveness of system engineers in handling open issues; and the numbers and types of change requests.

## 3.3 EXTERNAL BEHAVIOR GOALS

An important step toward system definition is to produce a statement of the target system's functional or behavioral requirements that meet certain criteria. It is quite natural to view the system and its environments as large cooperating processes that can be partitioned into successively smaller ones by functional decomposition. This approach, which is also used with real-time systems, focuses first on processes, then on control, and finally on data items. The results are good, provided the processes allow straightforward decomposition and the personnel are experienced in defining interfaces between the subprocesses.

To improve results without undue interference, certain guides for constructing functional requirements statements have been devised. As a point of departure, it is presumed the requirements would be *written* in natural language sentences. The guidelines are as follows:

- *Implementation Independence:* Each statement should be implementation free; that is, it should specify what is required of the target system but not how that requirement is to be met.

- *Unifunctionality:* Each statement should describe a single function (not multiple functions) to be featured in the target system.

- *Common Conceptual Level:* All requirement statements should be at the same level of abstraction to the extent possible.

Of the three, probably the most important characteristic is implementation independence. When faced with a complex design problem, there is a tendency for the designer to break the problem into subproblems that resemble familiar problems from past practice. Such a decomposition may well be a poor one from the system architecture viewpoint. To avoid this, it is important to exclude statements that primarily express implementation mechanisms.

System functional requirements are not usually expressed in this way in practice. The usual format is extended prose, including a mix of functional and procedural information spanning a wide range of conceptual levels. Consequently, it is necessary to have a way of translating the prose specifications into an appropriate form.

Once a satisfactory set of requirements has been generated, the designer must examine each pair of statements in turn and make a judgment as to whether a significant interaction exists. This may be done by considering how each requirement might be implemented in the target system and whether substantial interaction (interference or support) is likely. This is called interdependency analysis; the results are used to create a weighted interaction matrix.

The problem data may also be used to construct a nondirected graph, in which each node corresponds to a requirement statement and each link corresponds to an interdependency. The problem of partitioning complex systems into interconnected subsystems then reduces to consideration of simple connectivity properties of the graph obtained. The coupling variables correspond to the articulation set, while the simply connected components of the subgraphs formed by suppressing the articulation set correspond to the subsystems. For higher order graphs, the choice of articulation sets (and thus of decompositions) may be large and a choice must be made using some criterion such as minimal interactions.

### 3.4 SUMMARY

None of the elements of a user-centered design process are actually new. Instead, what is different is the emphasis placed on finding out from the users themselves what they want to do and what tools they need and prefer to use. User-centered design is a process that permeates the creation of a product or system. From the start, it draws on a wide variety of techniques to define systems based on objectively measured and verifiable user needs.

## 4.0 FUNCTIONAL ANALYSIS

This section considers the functional analysis step, the intent of which is to identify operating concepts and functional performance characteristics that can meet user needs for the system.

## 4.1 PROCESS-ORIENTED APPROACH

There are two approaches to design of reliable systems. A process-oriented approach, viewing computation in terms of procedures acting on data, is traditional. In this view, resources (data and devices) are bound to the address space of a process during its execution. The process is responsible for recovering its locus of execution, along with the resources bound to its address space. Checkpointing and rollback serve as the primary recovery mechanisms for constructing resilient processes. The possibility that rollback of one process may lead to a cascade of rollbacks then becomes a key concern. The main alternative is an object-oriented viewpoint, in which computation is viewed in terms of objects acting on messages. The process-oriented approach is considered here; the object-oriented approach is addressed in the section that follows.

A combat system can be viewed as a system for processing targets, as illustrated in Figure 5 above. In essence, each warfighting path then constitutes a sequential process for end-to-end engagement of a target. Section 2.0 of Reference 10 provides a conceptual and generic description of what a combat system is and does in terms of basic warfighting processes. The point of departure for a process-oriented approach to combat system architecture design is thus a high-level definition of the engagement processes to be conducted. What is necessary is to understand the basic purpose of each process and how it operates. This involves identification of all process boundaries, work groups, outputs and customers, inputs and suppliers, subprocesses and flows. The procedure given below is based on work by an AT&T Quality Steering Committee (Reference 11).

*Preparation:* First, draw a box to represent the process. The box itself represents process external boundaries. Inputs from suppliers enter the process box from the left, and customer outputs exit to the right, across the boundaries.

*Step 1:* List major work groups, dividing the process box into columns and using one column for each work group involved. Depending on the level of the process, these work groups may be organizational units, functions, departments, or teams intrinsic to process operation. Unless you are working with a very high-level process, a diagram with more than five groups is probably too detailed. The groups represent internal process boundaries.

*Step 2:* Identify process outputs and their recipients, who are viewed as the customers served. These are listed to the right of the process box.

*Step 3:* Identify all process inputs and suppliers, listing them to the left of the process box.

*Step 4:* Identify subprocesses and flows. One process input is taken at a time. The objective is to identify what work activities the input feeds, what group performs those activities, and what outputs are produced. Activities should not be defined so broadly as to cross over between groups (columns). Within columns, if more than three activities appear in a series without any external inputs, the diagram is probably too detailed. For each activity thus identified, proceed to identify the downstream activities that are fed by the outputs produced. This step is iterated until all the process inputs and outputs identified in Steps 2 and 3 are connected. The connecting paths consist of a series of work inputs, activities, outputs, and information flows. Details should be avoided; the aim is to produce a broad description for the overall process.

*Step 5:* Validate process definition by checking the flow diagrams produced against how the process really operates. Particular care must be taken to identify rework loops, ad hoc procedures, and any workarounds that may exist. Usually it takes several passes to get the diagram right. For a complex process, it may be useful to first list the activities of each major group and then consider the interconnections.

Additional steps may be employed to rationalize the process (i.e., to simplify its operation) or to examine its behavior at interfaces. The idea is to consider the following questions for each activity.

#### 4.1.1 Process Simplification

- Is the activity needed? Does it add value?
- Is the activity performed to accommodate errors–e.g, rework?
- Is the activity performed to undo the work of someone else?
- What opportunities for creating errors are introduced by rework?
- What are the obvious redundancies?
- Should someone else perform the work activity?
- Should the activity be combined with other activities?
- Should activities be run in parallel instead of series fashion?

#### 4.1.2 Behavior At Interfaces

- How do things get lost, changed, or misinterpreted between activities?
- Is there adequate feedback/communication between activities?
- Are clear internal customer/supplier requirements established?
- Are roles and responsibilities clearly defined?
- Are there undue delays?
- What practice determines the order in which work is processed, including special cases such as work-arounds or ad hoc procedures?
- Is there a more efficient or effective way of transmitting information or materials?

Sources of error, potential bottlenecks, and the adequacy of internal controls, including management of process change, must also be considered. Where possible, it is important to interview the people who do the work to obtain information on internal process problems. This approach is based on principles of task decomposition and has been tested by long use.

### 4.2 STATE SPACE REPRESENTATION

Concurrently with the functional analysis, a variety of system engineering studies are performed to select from alternate choices of function sequences to determine the best system design approach and to make tradeoffs in the grouping and arrangement of functions to form a functional architecture for the combat system. Analytical studies are an important part of this design process. Usually, the following four steps are involved.

- Developing a model or representation of the physical system
- Developing a mathematical representation of the physical system
- Analysis to determine the general properties of the system and its responses due to input signals
- Design that is carried out on the model of the physical system to improve or optimize performance by adjusting system parameters

In conventional control problems, mathematical models form the basis of design techniques–that is, of analysis, decomposition, and control synthesis. Quality design work involves the specification of many variables that together define a product, how it is made, and how it behaves. Reference 12 shows how a structured approach can be used to organize the design of a system, to develop an effective engineering plan, to show where estimates are required, and to analyze the flow of information in design work.

The first step is to list the variables that define the design of the system. Each variable is then considered to see what other variables must be known before its value can be determined. This implies a precedence ordering of variables, and the predecessors for each variable must be identified. This defines an interaction matrix **M** for the design problem. In effect, **M** defines required information flows for solution of the design problem. A mark in the ith row and jth column of **M** means that determination of variable $\mathbf{x}_i$ depends on $\mathbf{x}_j$. The diagonal elements are all marked and circled. If all entries above the diagonal are unmarked, the variables can be determined one at a time. If the ijth mark falls above the diagonal, however, the value of $\mathbf{x}_j$ must be found to begin an estimation process for $\mathbf{x}_i$.

To identify the structure present in the process data, it is often convenient to construct graphs using the following conventions:

- The nodes of the graph are the problem variables
- The arcs represent elementary processes (nonvalued)

The result is a graph representation for the input/output structure of the system of interest. This approach is often used to identify subsystem structure within a high-order system.

In general, some of the design variables are interdependent so that the interaction matrix **M** contains circuits. This makes it impossible to reorder the variables to that **M** is lower triangular. However, the variables can be reordered so that each mark appears either below the diagonal or within square blocks set on the diagonal. A matrix ordered this way is called block triangular.

Decomposition, reduction, and multicriterion optimization methods call for three things: knowledge of the system; a well-defined state space model; and where applicable, a definition of the subsystems and their interactions. Generally, opportunities to observe the complex system of interest are essential to gain the desired state of knowledge. In particular, two preliminary steps may be needed: analysis of system data, particularly to define interconnections and display the system's internal structure; and structural analysis to break down the system and identify subsystems.

## 4.3 ANALYSIS OF PROCESS DATA

A complex process often can be broken down into a collection of smaller, nearly independent, but interconnected subprocesses. The overall approach may be to assume an hierarchical structure exists and to study system data in order to define interconnections and display the system's internal structure. System identification is particularly difficult (and important) for command and control functions.

This may be accomplished through statistical methods (such as principal components analysis, regression, canonic analysis, or factor analysis), indexes of similarity and dissimilarity, and by an informational approach. The data consist of system information vectors for several different observations. Analysis may give a transition or coupling matrix, or equivalently some graphic representation of the complex system of interest.

A key special case is the technique of natural decomposition that is based on the presence of weakly coupled subsystems within the system. Natural structure is present because all the variables are not related to each other, or because there exists a hierarchy in the strength of the couplings between them. Analysis of subsystem stabilities and properties of the interconnections will then shed light on overall system stability and permit appropriate hierarchical or decentralized controls to be put in place. Such characteristics hold for a large class of interconnected systems found in practice. In addition, this technique allows full exploitation of prior knowledge about the complex system or process of interest.

## 4.4 CONTROL OBJECTIVES

Next, the task of formulating concise objectives for combat system control characteristics, based on general operational needs, must be addressed. Maintaining the user's perspective, attributes for system functionality, usability, performance, and cost are to be identified. Overall capability, environmental resistance, reliability, operating constraints, safety, affordability, and extensibility must be considered. Once corresponding quality measures and target values are established, success measures can be defined to help designers determine when to stop the iterative design process.

It is usually necessary to start with a qualitative formulation of control objectives for a given combat system. Two categories of objectives are considered. The first reflects operational feasibility concerns. Associated control objectives may originate in performance quality specifications, survivability or environmental considerations, and operational requirements. They are usually a function of interaction process variables that must be kept within specified bounds despite the uncontrolled entry of disturbances into the system.

The second category is derived from economic considerations. These enter the picture only if, after satisfying the first category of objectives, opportunities exist to manipulate the system into one or more relatively economical operating regimes. Feedback loops can sometimes be formed to govern economic performance.

## 4.5 MEASURED AND MANIPULATED VARIABLES

Most available control theories assume that measured and manipulated variables are selected prior to control design. One of the basic questions a system engineer faces in designing large and complex systems thus goes unanswered. This section considers the selection of measured and manipulated variables. Reference 3 gives a systematic procedure for making these choices in place of the rules of thumb normally employed.

Manipulated variables are the inputs adjusted by controllers to compensate for the effects of disturbances. It is not necessary that there be a one-to-one correspondence between manipulations and process variables, since various combinations of the latter sometimes can be used. It is important to identify all available feasible manipulations, both primary and secondary. The particular set chosen for use in control design will drive response capabilities of the system and influence its ability to achieve the control objectives on a continuous basis. Reliability and ease of operation are important as criteria in the selection of manipulated variables.

A set of measured variables is used for system monitoring and control purposes, forming the basis for synthesis of a proper plant monitoring system. Operability objectives usually directly dictate the measurements that should be made for monitoring system operation. For economic objectives, the variables of interest often cannot be measured directly, and so appropriate secondary variables must be selected for measurement. Both primary and secondary measurements must be identified for use in direct and indirect (inferential or adaptive) control loops.

The disturbances that the system will experience during operational use must also be considered and their impact on performance evaluated before a satisfactory control structure can be determined. The term *disturbance* is borrowed from the field of adaptive control, where output feedback is used to bring the controlled plant into conformity with a reference model specifying desired performance. However, noise present in output measurements can interfere with error assessment. Since a plant is rarely linear over its entire operating range, similar effects may occur due to modelling error. In addition, plant parameters can vary with time, as such variations provide the rationale for adaptive control. In all these cases, the output of the plant can be expressed as the sum of two components, one of which may be considered a disturbance for the purposes of analysis. A key design objective is to assure the boundedness of the output error between plant and model.

By shifting to an external or implicit reference model, it is possible to regard the disturbance vector as simply the vector of external inputs to the controlled process. Just as control objectives can be divided into layers (self-organizing, adaptation, optimization, and regulation), so the disturbance vector can be partitioned into several components. For instance, the disturbance vector is often partitioned into stationary and nonstationary components. The former correspond to fast variations that must be suppressed by reactive or reflexive methods, and the latter to persistent and/or periodic variations that must be handled at higher levels in the organization for battle. In general, a corresponding set of disturbances will be identified for each distinct set of control objectives present in the system.

Two forms of engagement interaction can occur. A reactive engagement is one in which some target is treated as a disturbance input requiring an event-oriented or transaction-processing response. The second form involves a transformational process that, like decentralized execution of STW, may involve a batch processing mode. To survey and then classify the characteristics of possible disturbances is one of the first steps in developing control structures. it helps to decompose the control task into its regulatory and optimizing parts, and it also determines the extent of each task. Further, the operational impact of the various slow disturbances entering the system drives the extent of the optimizing control structure and provides the initial guidelines for its design.

## 4.6 PERSPECTIVE

Defining high-level functional requirements for the system of interest and partitioning them into subsets in an effective manner may be called system architecture design. Generation of such a design is the aim of the functional analysis step in the conceptual design process. Results from this stage of design may be used to prepare functional flow diagrams and descriptions ($F^2D^2$) conforming to MIL-STD-490, which sets forth practices for the preparation and interpretation of military specifications. The $F^2D^2$ approach involves a top-down functional analysis, with increasingly more detailed information arranged in tiers. The different tiers never appear on a single diagram, each giving a self-contained and complete description of system functionality at a chosen level of detail. The tiers are structured as follows:

*Tier 0:* System requirements are identified and translated into functional requirements.

*Tier 1:* A top-level (Tier 0) diagram is analyzed and partitioned into subsystems. Each block shown in the top-level diagram is analyzed and translated into design requirements for subsystem functions. At this level, implementation of the functions is not addressed.

*Tier 2:* Required functions of the subsystems are analyzed and then partitioned into requirements for major equipments, manned operating stations, and computer programs.

*Tier 3:* Required functions of major equipments, manned operating stations, and computer programs are analyzed and then partitioned into requirements for (1) cabinet drawers and consoles, (2) operator tasks, and (3) computer program modules.

Most likely, only the first two tiers will be needed for the functional analysis phase of conceptual design. Important aspects of the Tier 3 analysis may be addressed in requirements allocation using object-oriented design methods, which are discussed in Section 5.0.

Creation of large, complex systems is very difficult and challenging work. It is difficult for system designers to mentally grasp all, or even most, of a system's parts at one time. Structured design techniques are widely used to overcome this obstacle. Reference 13 outlines an overall structured design method as follows:

1. Develop a method-independent and complete description of the external behavior required of the system.

2. Choose an appropriate organizational framework for the system.

3. Proceed to decompose overall functions into subfunctions, and compose primitive functions into higher level functions until all the requirements have been satisified.

4. Using various design rules and measures of coupling and cohesion, refine the design for increased modularity, extensibility, and likely reusability of modules.

5. Complete detailed designs as necessary for all modules.

In general, system partitioning into modules, module arrangement, and definition of interfaces are not sufficient to guarantee real-time reliability. Methods based on the notion of finite state machines may be used to examine key timing constraints and decision structure factors important in design of time-critical systems. Users need to know not only what the black box will do, but also how long it will take and what it will do with unanticipated problems.

# 5.0 ALLOCATION OF REQUIREMENTS

This section addresses the requirements allocation step of conceptual design. This step allocates system performance, integration, and affordability requirements on a functional basis among the subsystems identified by functional analysis. Section 5.1 thus considers how the problem of allocating functions between men and machines can be addressed. Since control structures must support the battle organization, combat system design efforts should be grounded in a clear statement of the role of humans. Section 5.2 considers the object-oriented approach to structured design, which provides a systematic framework for allocation of requirements to subsystems (objects).

## 5.1 ROLE OF HUMANS

Recent research (for example, Reference 14) suggests that once the main operational objectives have been identified, the roles that humans will play in achieving these objectives should be the first concern of system design. For this reason, it is essential to develop a clear understanding of the tasks to be performed by the potential users. Ways of assisting humans in these tasks can then be considered. In particular, allocation of basic weapon control tasks to men vs. machines is a primary issue for design. Reference 15 considers rules for allocating functions to human vs. machines. Basic design principles include the following:

- *Mandatory Allocation:* Human control may be required because of a *role of humans* statement developed by human factors specialists as a cornerstone of the system operating concept. Legal, safety, and performance considerations can also drive the allocation.

- *Balance of Value:* A hypothetical allocation can be based on estimates of the relative value of humans or machines as performers of the intended function.

- *Utilitarian and Cost-based Allocation:* A function may be allocated to humans simply because they are present and there is no reason why they should not perform the work. Otherwise, allocate on the basis of least cost.

- *Affective or Cognitive Support:* Affective support refers to the emotional needs of humans, such as their need to know their work is valued, to feel personally secure, and to feel they are in control. Cognitive support refers to the human need for information to be ready for actions or decisions that may be required.

Information about relative goodness of men or machines for a given task is often capsulated in the form of a Fitts list, sometimes called a Men Are Better At–Machines Are Better At (MABA-MABA) chart. Several more detailed procedural guides are available, but have not become widely used tools of system design.

## 5.2 OBJECT-ORIENTED METHODS

In object-oriented design, computation is viewed in terms of objects acting on messages, and each resource is permanently bound to the address space of an associated process–the object manager. Every other process that wants to access this resource does so by invoking the interface procedures supported by the object manager. Each object manager virtually defines an error-confining domain within which it is responsible for enforcing needed concurrency control rules and recovering objects from faults and system crashes. The primary recovery mechanisms include forward/backward logs, careful replacement, and object replication. Processes are no longer held responsible for recovering the objects they access during their execution; however, they are still held responsible for recovering their execution locus. This requires establishing recovery points and rolling back a process to some recovery point. A major advantage of the object-oriented approach is the clean separation between the recovery functions for processes and objects. Another advantage is that for each object type, recovery mechanisms can be tailored to the type integrity requirements.

When object-oriented methods are used for structuring, the system and its environment are viewed as a collection of interacting objects. Objects interact through well-defined interfaces. Since the internal structure of one object is not directly accessible to another, each object represents an independent domain. The whole system thus can be viewed as a collection of objects. From the viewpoint of reliable system design, such an approach is very attractive because it supports confinement of errors within an object boundary. The following general design principles are used for identifying, relating, structuring, and controlling system objects.

### 5.2.1 Identification of Objects, Functions, and Services

Postulate a set of objects and then associate them with major functions. Named objects should be used to represent system and software entities, thereby creating a representation for essential resources and processes and permitting the latest possible binding of objects to physical resources. The problem can then be decomposed into parts as follows:

- What objects are required:
- What should they be named?

- How should they be located and addressed?
- How should they communicate?
- How should they be controlled and coordinated?

Ideas of class and inheritance are used to guide definition of objects. Classes are based on a template giving a general description for the purpose, structure, behavior, and interfaces of associated entities. They are related by inheritance rules, usually producing some form of hierarchical relationship that encourages modularity in the system. Objects are defined as instances of the classes and communicate by exchanging messages. Objects uniquely encapsulate a system's design both in terms of its data structures and in terms of required behavior, which helps to ensure a tight binding between system definition and implementation.

5.2.2 Identification of Functional Modules

Specify objects and functions by services performed. This is equivalent to specifying a virtual machine as the vehicle for achieving system requirements. The chief advantage of this approach is that concerns about hardware or software details (e.g., bus structure or network protocols) are not allowed to interfere with the process of identifying desirable system properties, which should be largely independent of implementation methods. This helps to separate software architecture from software design.

- Design for multiple use modules by specifying general functional modules for all nonunique application processing. This helps (1) to reduce redundancy of effort in module development, (2) to reduce redundancy of storage space and execution time during network operation, and (3) to confine the effects of component changes to a small number of standardized modules.

- Make the modules as independent as possible by allocating a single major function or subfunction to each module.

- Unique application functions should be put in application modules.

- Use system state information as soon as it is available in order to eliminate the module coupling that would be involved if the system were designed to recapture the information later. State information should be used to plan resource use as far as possible in advance, avoiding unnecessary message exchange and module interaction.

5.2.3 Naming and Addressing of Objects

Make objects accessible by the type of service performed (or subject), independent of location. Use location-independent means of access to objects so that users are free of constraints in their access to resources by host identity or geographic location. Instead, users are free to access resources by type of service. (This can be done by logical bus communications on local networks.)

5.2.4 Interprocess Communication

- Hide knowledge of message communication from objects. This is accomplished by not storing name, address, function, or other key information within objects. Instead, one object invokes another indirectly by using service codes and a network services directory.

- Provide direct communication between objects. A method of maximizing connectivity and accessibility is to use a logical bus communication system for local area networks (LANs). Routing through intermediate objects introduces queueing delays and will make maintenance difficult as the network configuration changes in the future.

- Separate control data from user data. A feature of the logical bus is the separation of control data from user data by using separate logical data links and ports for each. This aids maintenance, since changes in control data procedures and message formats will not affect user data protocols and vice versa.

- Minimize control message exchange between objects. An alternative to message communication is the remote procedure call. This method appears to require more explicit binding between objects than does message processing. Interrupt-driven message recognition should be used instead.

5.3 SUMMARY

As compared to the classical approach to structured design, the object-oriented approach is an important step forward. It raises the problem of allocating functions to objects to new levels of visibility and effectively communicates a potential for continual improvement of system performance and cost characteristics through attention to this phase of design. See Figure 19 in Section 7.1.3 of this report for an illustration of the potential significance of an object-oriented approach to combat system engineering.

## 6.0 PRELIMINARY SYSTEMS ANALYSIS

This section outlines the modelling and analysis techniques that comprise the foundations of control theory and are needed for use at the fourth step in conceptual design–preliminary systems analysis. At this stage, alternative design solutions are identified and screened to eliminate those that are clearly unattractive, leaving only the most promising for evaluation. The work is performed in an iterative process and must cover an array of tradeoffs adequate to support the major design decisions that are to be made.

In this report, a plain font is used with lower case letters to denote scalars, while upper case letters are used to denote sets. For example, x and t denote scalar variables, while m and n are positive integers. Greek letters are used at times for key parameters. Such letters as G, S, and T are used to denote graphs, sets, and transformations as indicated in the text. However, letters M and N are used to denote the limits of an index set. A bold font is used with lower case letters to denote vectors; and with upper case letters to denote matrixes. Thus **x** denotes an nx1 state vector and **x**(t) is a vector valued function of time. Similarly, **u** denotes an mx1 input vector. The symbols **A**, **B** denote matrixes of dimensions nxn and mxn, respectively.

## 6.1 LARGE-SCALE, COMPLEX SYSTEMS

In the design of large and complex systems, great difficulties may be encountered in developing mathematical representations of physical systems and in conducting analysis and design efforts.

Large-scale systems are usually defined as those whose mathematical representations contain 100 or more state variables. However, these representations often contain linear equations or can be treated using normal, traditional mathematical tools. A spatial or geographic distribution is often associated with large-scale systems. Examples include large production units and distribution and service networks such as the Navy logistical support system. Difficulties arise in developing mathematical representations for such systems due to their size and the information flow constraints imposed by geographical distribution.

Complex systems, on the other hand, have representations given by systems of partial differential equations, highly nonlinear models, and qualitative representations using *fuzzy* concepts. Complex systems often have a large number of variables with many links and interactions between them. Great difficulty is therefore encountered in representing them with traditional mathematical tools. Usable models are difficult to develop, and those developed usually have a high degree of mathematical complexity.

Although there is no universal definition for large-scale and complex systems, Reference 16 indicates that they often have the following characteristics:

- Multiple controllers or decision makers are present, and the control computing is decentralized to a significant extent.
- Controllers have different but correlated information available to them, possibly at different times.
- Actions taken by controllers at one level are being coordinated at another level in a hierarchical (multilevel) structure.
- Controllers may operate as a team or in a conflicting manner with multiple or even conflicting objectives.
- The system must be represented by imprecise *aggregate* models.
- Satisfactory control may be achievable by means of suboptimal or near-optimum controls, sometimes called a *satisficing* strategy.
- Centralized control methods cannot be used due either to a lack of centralized computing capability or lack of centralized information.

Complex systems involve significant difficulties in problem analysis, decomposition, aggregation, and control. The analysis phase calls for definition of the I/O, controls, construction of the m del, estimation of the parameters, and definition of the criterion. This phase may not yield to c rect attack in cases that involve high-order systems. Many large-scale systems found in practical applications are not linear and involve parameters that are unknown or perhaps imprecisely modeled. A common practice is thus to design for operation around a set of nominal operating points or trajectories. Even so, the models formulated may be too complex and too detailed for immediate application of optimal control techniques. Measures must then be taken to simplify the overall problem or break it into subproblems that are easier to solve. These problem characteristics are summarized in Figure 8.

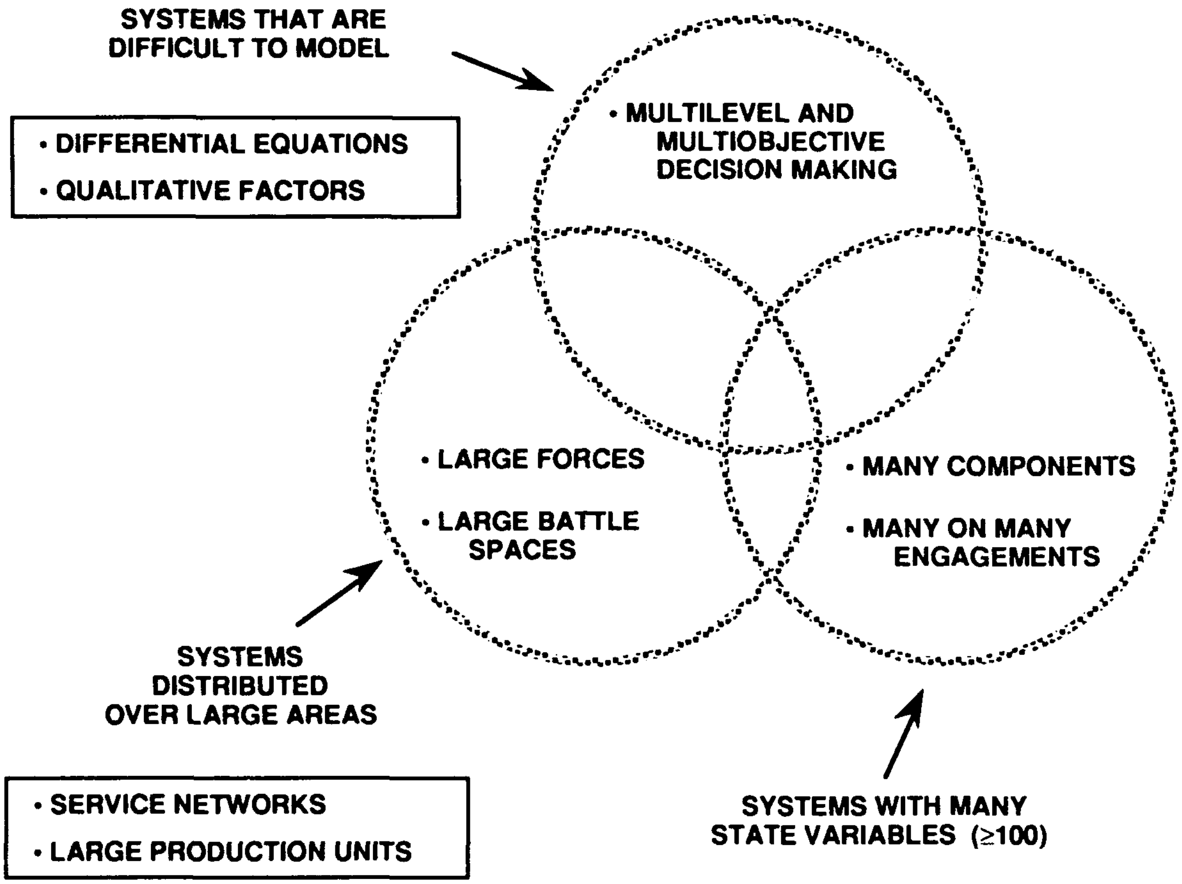



FIGURE 8. PROBLEM CHARACTERISTICS

Combat systems of the World War II era were a federation of loosely coupled and coordinated weapon systems. Each weapon system was typically configured with one engagement control loop. Since then, with the advent of computer technology, the trend has been toward more weapon system automation and control and more coordination among weapon systems. Orthogonal to this trend has been the trend toward more cooperation or sharing of information among combat system elements. This has greatly increased the linkages and interactions within combat systems. These trends are likely to continue.

What makes the combat systems of today complex is the high degree of coordination and information exchange among the system elements–from the human decision makers and system managers to the electronic loops controlling the various system processes. Extension of this trend to systems with elements distributed among several different ships will further increase complexity and scale.

Advancing technologies for automation and control have made the engineering of large-scale systems an area of significant and growing practical importance. Decomposition, aggregation, and model reduction are the principal techniques used to deal with the associated complex modelling problems and have received considerable attention over the last 15 yr. Model reduction methods transform a complex problem into one of reduced scale or dimension to which standard methods (e.g., mathematical programming) can be applied.

## 6.2 DECOMPOSITION METHODS

The partitioning of a combat system is not dictated by computational considerations but is part of the overall design strategy. In fact, an inappropriate decomposition can add to problem difficulty.

The system engineer should first seek to place the control objectives in some hierarchical order, so that corresponding control systems can be synthesized in a sequential manner. The next concern should be the possibility of decomposing the overall system in such a way that only smaller problems need to be addressed. Of course, the potential impact of any decomposition on command and control must also be considered. Both concerns must be reflected in formulation of a criterion for decomposition. The main guidelines are as follows:

- Form subsystems around a common functional goal.
- Provide for minimal interaction among subsystems in the final configuration; i.e., an involved coordination among subsystems should not be required every time a disturbance enters the system.

### 6.2.1 Horizontal Decomposition

Large and complex processes often can be decomposed into a collection of smaller, nearly independent, but interconnected subprocesses. The separate pieces have reduced dimensionality and are thus easier to solve than the original problem. Each subsystem is then handled by a local control unit, and the overall process may or may not be coordinated by higher levels of control (hierarchical or decentralized structure). Without the natural structure exploited by this approach, control actions taken in one subsystem might require simultaneous and compensatory actions elsewhere, making coordination difficult. The advantages of decomposition can be realized despite the presence of weak interactions between subsystems, but the overall process must be separable to at least a first approximation. System horizontal structure generally reflects temporal, functional, or spatial connectivity of the complex system or process of interest.

While decomposition is easy in theory, the subsystem structure must, in practice, preserve constraints, information structures, and authority structures that are societally acceptable. Success is determined, in part, by the particular decomposition and coordination techniques employed. Three main principles are used in coordination, as described below.

1. *Balance Coordination:* Each subsystem treats the model coordination variable as a pseudocontrol variable in solving its own subproblem, while the

coordinator upgrades and supplies each subsystem with its goal coordination variables.

2. *Prediction Coordination:* The coordinator predicts the model and goal coordination variables of each subsystem and then supplies these values to the subsystems.

3. *Penalty Coordination:* The coordinator upgrades and supplies model and goal coordination variables to each subsystem, where coordination is achieved by the penalty weighting of coordination errors.

Horizontal decomposition is associated with a general mathematical formalization, unlike the case of vertical decomposition, which tends to be driven by decision and control capabilities rather than process characteristics. Briefly, vertical division is difficult to formalize on the mathematical level, the chief concern being to compromise between the period of intervention at one level $T_i$ and the sophistication of the algorithm employed at this level.

6.2.2 Vertical Decomposition

As a complement to horizontal decomposition methods, this approach deals with the complexity of the overall control response function by breaking it down into several functional or temporal layers of control. For example, the process may involve layers with different response dynamics and perhaps different timescales. This leads to vertical and temporal decomposition approaches intended to shape a satisfactory response function by the coordinated action of several simpler controllers. The overall approach is one of successive approximation, in which initial solutions for the local controllers are simplified by relaxation of the coupling constraints.

The layers of the hierarchy represent different kinds of control functions, and so require different kinds of computation and information processing algorithms. Figure 9 shows a multilayer control structure based on a functional decomposition of a complex control problem. The structure shown is widely employed in the field of automatic control.

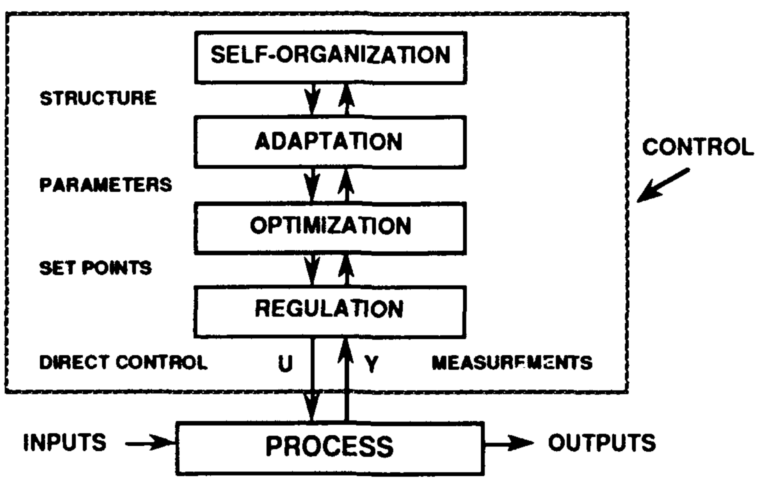



FIGURE 9. MULTILAYER CONTROL STRUCTURE

- *Regulation or direct process control:* An important characteristic of the first layer is its ability to interact directly with the controlled plant and on the same timescale. Data acquisition, event monitoring, and implementation of essential control actions are primary subfunctions.

- *Supervisory:* Another layer provides static or dynamic optimization of set points or input trajectories for the controlled plant. This defines the immediate target or task to be implemented by the first layer. In general, there may be a number of operating modes or topologies identified for the system. Each may then have a different mathematical model through which information describing the current state of the system is transformed into directives applied to the first layer function. In particular, it provides for transition of the plant from one operating mode to another.

- *Adaptation:* This layer may intervene in the operation of the lower levels to modify the process model or control law to be implemented. Typically, the actions taken at this level reflect operating experience over a period of time. The actions are discrete, occurring at predetermined time intervals or in response to certain events (such as operator inputs).

- *Organization:* This layer provides for choice of model, control, or policy structures in terms of the environment. It may intervene directly in operations of the lower control layers by a mode selection mechanism. More typically, it may be implemented through an update of the control system design and reprogramming of control computers.

Generally the levels are ordered by timescale, degree of aggregation, frequency of control, or other attributes that must be considered at the design stage. For example, if $t_i$ denotes the period of intervention at level i, then $t_1 < t_2 < \ldots . < t_n$. The control structure thus acts as a selective filter of the different disturbances affecting the process, the most rapid response measures occurring at the lower levels. The structure also plays a role in organizing the flow of information through the system and providing mechanisms for the effective use of feedbacks for control and decision making.

The different layers represent different kinds of control functions, and so require different kinds of computing and information processing algorithms. System integration is based on a clear-cut assignment of tasks and responsibilities to the different layers of control.

A second form of multilayer control structure, based on a temporal decomposition of the control problem, is also widely used. In this approach, the control or decision-making problem is partitioned into subproblems based on different timescales relevant to the essential action functions. These timescales reflect such factors as delays to obtain information or respond to prior actions, temporal characteristics for different types of disturbance inputs, and the changing value of delayed information or actions.

Vertical decomposition may be used together with horizontal division if the process is structurally complex. This leads to multilevel, multiobjective structures. The direct control level might be decentralized into N local control units, for example, while the optimization level is broken down into two sublevels, one providing N different local controllers for optimization of the direct control level and a second providing coordination. Vertical decomposition on the basis of time (planning horizon) is often used in production management, an application with characteristics different from the usual control engineering application.

### 6.2.3 Spatio-Temporal Decomposition

This involves a combination of the two previous types of decomposition, which is of interest for geographically distributed systems and gives rise simultaneously to different dynamics.

### 6.2.4 Decomposition Quality

One criterion for a good decomposition is that it consist of modules that possess high strength (internal binding) and are also weakly interconnected. The strength/coupling criterion may be quantified in the following way. Suppose the graph representation of the target system design problem is decomposed into nonoverlapping subgraphs $G_1,....,G_n$. If $s_i$ denotes the strength of subgraph $G_i$ and $c_{ij}$ the coupling between subgraphs $G_i$ and $G_j$, let

$$M = \sum_{1 \le i \le n} (s_i) - \sum_{1 \le i \le n-1} \sum_{1 \le i \le n+1} (c_{ij})$$

The index M may be used as a figure of merit for the decomposition. The most useful definitions for the $s_i$ and $c_{ij}$ discovered so far are as follows:

$$s_i = [l_i - (n_i - 1)]/[(n_i(n_i - 1)/2) - (n_i - 1)]\ [w_i/l_i]$$

$$c_{ij} = (l_{ij}/n_i n_j)(w_{ij}/l_{ij}) = w_{ij}/n_i n_j$$

where $l_i$ is the number of links in $G_i$, $l_{ij}$ is the number of links between $G_i$ and $G_j$, $n_i$ is the number of nodes in $G_i$, $w_i$ is the sum of weights on links in $G_i$, and $w_{ij}$ is the sum of link weights for links between $G_i$ and $G_j$. Note that $s_i$ and $c_{ij}$ are in [0,1]. Also, the $s_i$ terms are normalized with respect to the minimum connectedness of the corresponding subgraph. That is, $s_i$ measures the extent to which the strength of $G_i$ exceeds the minimum necessary to form a connected subgraph.

This approach is described in more detail by Reference 17. The problem of locating an optimal decomposition may be formulated as a nonlinear integer programming problem, but practical methods do not exist for exact solution. Partitioning methods and clustering methods, based on a similarity measure defined on pairs of nodes (i,j) are used to seek approximate solutions. Reference 18 indicates how decomposition quality measures can be used to design modular computer programs.

## 6.3 AGGREGATION

Large models are often needed to capture all available knowledge about a complex system. Despite the power of modern computers, it is often difficult to deal with such models. Generally one of two approaches may be employed. The first is problem decomposition, based on physical or mathematical considerations. This is the classical approach in hierarchical control theory. The second is to be more selective about the information used. In this approach, the aim is to replace the original representation by one of smaller size while keeping essential features of the problem.

Although the two approaches may appear to be quite different in principle, they may overlap if one considers systems where two or more timescales are present. In addition, aggregation techniques may be used in a hierarchical control approach to simplify one or all of the subsystems since the higher the level, the simpler the models have to be for efficient decision making.

The term aggregation, first used by economists, is widely used in its most general sense of approximation. An acceptable approximation to the I/O characteristics of a large-scale system is generally obtained if the dominant modes of the system are retained in the aggregated model. Several methods can be used to define the modes of interest.

## 6.4 PARTITIONING AND TEARING

Extensive decomposition may be necessary to reduce the scope that each designer must comprehend at one time. The two key questions here are how to measure the quality of a given decomposition and how to find the best decomposition (or one close to the best). Beyond questions of basic connectivity, however, the process of decomposition is not well known or understood. It is a process based almost entirely on personal judgment. which in turn is based mainly on past practice. No widely accepted measures or indexes of good preliminary decompositions exist. Nor have any explicit methodologies for decomposing large complex systems been previously explained or widely used.

Reference 19 presents some new ideas and techniques pertaining to the search for those "subsets of requirements that should be dealt with independently." The ideas are based on the centrality of a system's functional requirements and their interactions and on the need for a systematic approach to manage the complexity of large system design effectively. His approach was to search for subsets of system requirements that are closely related to each other but also relatively weakly related to other such subsets. This forms the principal objective function for requirements decomposition.

The blocks set on the diagonal in the block triangular form of the interaction matrix can be found by a procedure called partitioning and tearing. The technique was developed by Reference 20 for tearing large, sparse linear systems of algebraic equations into smaller systems and then assembling the partial solutions to form the solution of the overall problem. Here, the system of equations is considered to be the set of defining equations for the design variables. Partitioning can be accomplished for nonlinear as well as linear systems of equations without an explicit statement of the equations. The blocks produced correspond to the smallest sets of variables that must be determined jointly. The blocks are identified by tracing circuits; thus, all the variables associated with any circuit will be found in the same block. The process is called partitioning because it divides the variables into blocks, each representing a mutually interdependent subset. These blocks can be solved one at a time and the partition is unique. The algorithm presented is fast, and computing time does not appear to increase rapidly with the number of variables. Systems involving perhaps 50 to 100 variables can be solved without a computer.

It is sometimes important to obtain an ordering of variables within each block so that reasonable estimates can be made for marks above the diagonal. The aim, in fact, is to remove a set of marks from the block and reorder the remaining variables by partitioning so that the reordered block contains no marks above the diagonal. This signifies that although estimates must be obtained for the variables removed, no additional estimates are needed.

The marks removed are called *tears*. Essentially, a mark is chosen at which to tear each circuit of the interaction matrix. Variables in the associated block are reordered in a manner determined by the tear set. As long as blocks remain with more than one variable, circuits remain to be broken with additional tears. To choose a good tearing involves an interplay between analysis of interaction matrix structure and engineering judgment as to the why and how of interaction effects. The approach presented by Reference 20 was based on representing systems of equations as electrical networks. His choices for tearing were based on physical insights, and under such conditions the method has given excellent results. However, his work usually involved removal of enough elements that the pieces could be solved in any order. Here, only enough elements are torn to that the structural matrix can be put in block triangular form. References 21 and 22 identify procedures for use in tearing.

The resulting matrix, with variables reordered by partitioning and tearing, is called the design structure matrix. This matrix is useful in planning for the design work. Reference 22 recommends that the more difficult estimates be made by senior engineers, and that personnel working on variables in the same block be kept in close proximity.

## 6.5 STRUCTURAL ANALYSIS

### 6.5.1 Structural Controllability

Reference 23 shows that if a prior decomposition is not specified for the plant, designers can choose an appropriate decomposition on structural controllability considerations. Figure 10 shows the problem representations used for such analysis. Consider a time-invariant linear system with n state variables $\mathbf{x}_i$ and m inputs $\mathbf{u}_k$. The system is described by

$$\partial \mathbf{x}(t)/\partial t = \mathbf{A}\cdot\mathbf{x}(t) + \mathbf{B}\cdot\mathbf{u}(t)$$

which can be denoted [**A,B**] for convenience. Then the following definition can be given.

DEFINITION: A system [**A,B**] is said to be *controllable* if any given state vectors $\mathbf{x}_1$, $\mathbf{x}_2$ can be transformed into each other by some control input **u** in a finite time interval.

Controllability is equivalent to the following rank condition:

$$\textbf{Rank}\ [\ \mathbf{B}\ |\ \mathbf{AB}\ |\ \mathbf{A}^2\mathbf{B}\ |\ \ldots\ |\mathbf{A}^{n-1}\mathbf{B}\ ] = n$$

Many times, the entries of the matrixes **A** and **B** are not all known, so it is convenient to consider only which elements are zeroes and which are different from zero. Two systems [$\mathbf{A}_0$,$\mathbf{B}_0$] and [$\mathbf{A}_1$,$\mathbf{B}_1$] are said to have the same structure when they have the same zero elements. The system [**A,B**] has n(m+n) entries, and systems with the same structure form a subspace in the vector space $\Re^{n(m+n)}$. In addition, it turns out that systems of the same structure are either not controllable or their controllability is a typical property of that structure. Typical means here that controllability is expected with unit probability. This leads to the following definition.

DEFINITION: **[A,B]** is said to be *structurally controllable* if and only if any system of the same structure is controllable.

The concept of structural controllability, together with application of graph theory, leads to some interesting and useful results.

A graph representing **[A,B]** can be constructed as follows. For i=1,...,n let node $n_i$ correspond to the state variable $\mathbf{x}_i$. Also, for k=1,...,m let node $n_{n+k}$ correspond to the input variable $\mathbf{u}_k$. For any pair (i,k), an edge exists between nodes $n_k$ and $n_i$ or between nodes $n_{n+k}$ and $n_i$, if and only if the corresponding element $b_{ik}$ of **B** is nonzero. These edges are oriented (from $n_k$ or $n_{n+k}$ to $n_i$).

Reference 24 has established that structural controllability is equivalent to a graph structure that (1) contains no nodes that are not reachable from some input node and (2) contains no *dilations*. The concept of a dilation is described in Figure 10.

**MATHEMATICAL REPRESENTATION**

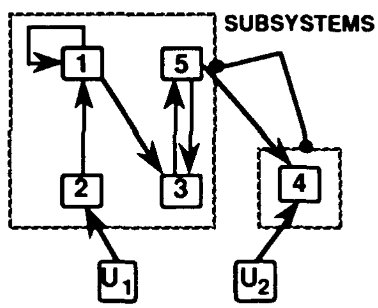
$$\dot{x}(t) = A\,x(t) + B\,u(t)$$

**SYSTEM MATRIX** / **CONTROLLER**

$$A = \begin{bmatrix} \bullet & \bullet & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ \bullet & 0 & 0 & 0 & \bullet \\ 0 & 0 & 0 & 0 & \bullet \\ 0 & 0 & \bullet & 0 & 0 \end{bmatrix} \qquad B = \begin{bmatrix} 0 & 0 \\ \bullet & 0 \\ 0 & 0 \\ 0 & \bullet \\ 0 & 0 \end{bmatrix}$$

- **SIMPLE, LINEAR SYSTEM**
- **CONTROL INPUTS $U = (U_1, U_2)$**

**GRAPH REPRESENTATION**




**CONTROLLABILITY REQUIRES:**

- **ALL NODES REACHABLE**
- **NO DILATIONS PRESENT**

FIGURE 10. STRUCTURAL ANALYSIS

DEFINITION: Suppose that for any subset S of state nodes, T(S) is the set of nodes from which an oriented edge goes into a node in S. A graph then has a *dilation* if some S has more nodes than T(S).

This would mean states in S cannot be driven independently by the inputs, thus making the system uncontrollable. The same research also produced a second set of equivalent conditions. Suppose we start with two simple graphs. The first, called a *stem*, contains only a series of linked nodes with no branches or loops. Note that a loop consists of a stem in which the first and last nodes are identical. The second type of simple graph, called a *bud*, contains such a loop, plus a single node outside the loop, and connected to it by a single arc. For systems with one input, Reference 24 has shown that a graph constructed of one stem, with perhaps several buds starting in nodes of the stem, yields a controllable system. This form of graph is called a *cactus*. It is possible to extend this result to multi-input systems. Thus a system **[A,B]** is structurally controllable if and only if the graph of **[A,B]** is a cactus or is spanned by a cactus–that is, can be transformed into a cactus by deleting some edges. By implication, every structurally controllable nxm pair **[A,B]** can be decomposed into m structurally controllable subsystems. This can be done in an algorithmic fashion, as shown by Reference 25, and must end in separated cacti. The algorithm is summarized as follows.

Starting with the structure graph for the overall system, edges will be deleted successively. At each step, the edge that belongs is deleted to a matrix entry of smallest absolute value and does not lead to unreachable nodes or to a dilation. The process of edge deletions continues until the graph is reduced to a number (≤m) of separated cacti. Each cactus represents a subsystem and the deleted edges describe their couplings.

A duality relationship exists between controllability and observability. Therefore, the above graph theoretical results are easily used to investigate a system's structural observability properties. It is necessary only to substitute the idea of output-unreachable nodes for that of input-unreachable nodes, and then to substitute the idea of contraction for that of dilation.

Systems that are both structurally controllable and observable are called structurally complete or structured systems. The combination of the results of structural controllability and observability shows that the structure graph G(**A**,**B**,**C**) of a system that is not structurally complete contains at least one input-unreachable or output-unreachable node, a dilation, or a contraction.

### 6.5.2 Consideration of Fixed Modes

Extending the concept of structural controllability to decentralized information structures involves another complication: i.e., systems that are structurally complete may yet have uncontrollable fixed modes, and it is necessary to get around their effects. (The notion of uncontrollable and unobservable modes in centralized control is generalized to the concept of fixed modes in decentralized control problems.) An eigenvalue $\lambda$ of **A** can be a fixed mode if, for some disjoint partition of information between two control units, $\lambda$ is at once uncontrollable from one unit and unobservable from the other. The fixed modes arising from decentralized control structures can be computed. The procedure is to find and compare the eigenvalues of **A** vs. **A+BKC**, with a block diagonal output feedback matrix **K** chosen at random.

It is important to recognize that in decentralized control systems, the control stations can communicate through the state space by the use of signalling strategies. In certain cases, fixed modes can be brought under control by such strategies. However, this leads to a nonlinear control law and an optimal state estimator that may not have finite dimension. This discovery led to much work aimed at identifying special cases where linear control laws are possible and/or where state estimation is separable from the computing of control inputs. This research showed that linear decision rules are achievable with partially nested information structures; see Reference 25, for example. The principle involved is that if action $\mathbf{u}_i(k)$ (by control unit i at step k) influences the information available later (at step k') to control unit j, then the latter should know by step k' whatever was known to unit i at step k. In particular, one step delay information sharing patterns yield both a linear control law and a separable estimator.

Available research reviewed by Reference 26 suggests that it is important for system designers to avoid fixed modes. In particular, each of the eigenvalues for the system matrix A should be controllable and observable from some control station. Where they cannot be avoided altogether, links should be provided between control stations to overcome the fixed modes by information sharing. For example, some target may be observable to subsystem **A** but *controllable* only to subsystem **B**. Information sharing between **A** and **B** may suffice to create an effective action path.

## 6.6 CONTROL QUALITY FACTORS

### 6.6.1 Research in Control Theory

Reference 26 reviews and evaluates the state of the art in design of decentralized control structures. Although such controllers had been designed and used for over two decades prior to this review, the design was based on ad hoc methods. More recently, there had been attempts to translate basic theoretical results on centralized controller design to problems of decentralized control. These studies identified a number of important theoretical and practical issues.

One of the most useful results in the theory of control for centralized systems is the certainty equivalence property. The significance of this property is that for linear dynamical systems acted upon by white noise disturbances, it is possible to design a controller that minimizes the expected value of a quadratic cost function by *separately* solving two deterministic design problems, one giving an optimal controller and the other an optimal state estimator. In decentralized control, constraints on the information structure do not allow for optimal solutions with separation between state estimation and the design of control laws.

A lot of research was done in the decade between 1974 and 1983 to identify special cases where linear solutions are possible and/or where a separation between state estimation and computation of the control inputs is maintained. For example, Reference 25 found that partially nested information structures, such as a one-step delay pattern for information sharing, lead to decision rules that are linear.

A factor in the breakdown of the certainty equivalence property is that one control station can transmit information to others through the state space by using a signalling strategy. This can require the use of time varying and/or nonlinear control strategies. Fixed structure linear controllers remain useful in many problems of practical interest, but decentralized observation or filtering may be necessary.

### 6.6.2 Performance Measures

Traditionally, performance of real-time control computers has been analyzed separately from that of the corresponding controlled processes. Performance measures used for real-time control systems were usually adapted from those devised for more conventional computers. However, there is a considerable mismatch between real-time control problems and conventional computing applications. Thus, control engineers are developing new procedures for use in specifying and evaluating controller performance. The idea is to find methods that can be systematically applied and will provide objective results that lend themselves to formal validation.

To remedy this situation, vector measures made up of such traditional indexes as reliability, throughput, survivability, or availability may be used. A real valued function of the vector then permits comparison of different vectors. One way around the usual difficulties is to express performance objectively in terms of the response time of the computer controller. From the point of view of the controlled process, the computer controlling it is a black box whose behavior is exemplified by its response time and reliability. The response time is a random variable, a function of current system state, system failure rate, and interference effects (electrical or magnetic), among other parameters. Control overhead is a monotonically nondecreasing function of response time, and a catastrophic failure occurs if it exceeds a corresponding hard deadline for the system.

### 6.6.3 Forms of Control Degradation

From the viewpoint of the controlled process, the control computer is a black box whose behavior is exemplified by its response time and reliability. If the controller's response is too slow, a catastrophic failure event (dynamic failure) will result. However, a variety of other process characteristics (control overhead, disturbances, unmeasured variables, constraints, and interaction effects) are also of interest in assessing controller performance. Abnormal or degraded control can result from the following:

- Controller passes incorrect output to actuator.
- Controller execution time is greater than nominal values (but less than demanded system change/update rate).
- Controller execution time is excessive, causing abortion of the execution sequence and generation of a new sequence.
- Information loss due to excessive loading of system elements.
- Control degradation by enemy command, control, communication, and intelligence (C3I) countermeasures.
- Loss of connectivity among system elements.
- Loss of synchronization among system elements.
- Uncertainty about positions and intentions of own force elements.

### 6.6.4 Design for Survivability

Reference 26 identifies possible sources of control failure as follows:

- Actual behavior inconsistent with the plant model used in design.

- Plant behavior changing with time, either slowly or abruptly.
- Failure of needed sensors and actuators to work properly.
- Loss of needed control information links.
- Partial failure of the controller itself.

Combat systems must be designed to operate against a wide range of threats in very difficult environments. With such high uncertainty, open loop control techniques simply cannot support adequate levels of performance; feedback control methods are needed. The feedback systems must, in addition, be highly reliable and fault tolerant. In practical applications, there can be little hope of designing control structures that can tolerate all the possible failures. It must be acknowledged that key components of the overall system will fail at times, and that such failures can degrade stability or performance. Essential protective measures thus include the following:

- Reliable design for normal operations.
- Hardening against selected classes of failure events.
- Failure detection circuits and algorithms.
- Procedures for reconfiguring the control structure for effective use of whatever resources remain in the event of serious damage.
- Standby and reserve components.

Procedures for multiobjective communications network design are given by Reference 27. The designers of communication networks often build a particular network configuration around specific processing, performance, and/or cost requirements, with little consideration of its stability under the pressure of link and/or node losses. This can lead to unidentified weak points in the network's basic node and link structure.

Effective design of a survivable communication network requires a means of accurately gauging the connectivity level of its interconnection structure as a whole. The Node Connectivity Factor (NCF) and Link Connectivity Factor (LCF) form global connectivity measures that are based upon classical graph theory involving both tree diagrams and graph structures. The NCF represents the physical stability of the network in terms of the average number of topologically critical nodes that must fail in order to force the remaining nodes into a standalone configuration. Similarly, the LCF is representative of the network's electronic stability as defined by the average contribution of each link to maintaining a minimally connected configuration. Both of these global measures reflect a worst-case analysis of the global connectivity of the network.

Also needed is a means of identifying how critical individual node and link resources are to maintaining the physical integrity of the network. These individual node and link criticalities are quantified by Node Decomposition (ND) and Link Tree (LT) indexes. ND represents a node's contribution to the physical stability of the network and ranges in value between 0 and 1. The higher ND is, the more critical the node. LT represents a link's contribution to the electronic stability of the network and also ranges in value between 0 and 1. The most survivable network is one in which all nodes and links appear to be equally critical.

One popular algorithm used in current packet switched communication network design procedures is known at the cut-saturation algorithm. Reference 27 modifies a version of this algorithm to include survivability considerations in making decisions about adding or deleting links to meet specified throughput goals. The approach can be outlined as follows.

- A single set of operating conditions (topology, traffic loading, and delay) was postulated.

- Link use, link tree index, node decomposition index, link distance, and average hop distance factors were introduced as design criteria. In essence, the information provided by the ND and LT indexes is used to indicate which links to add or delete. If a link must be dropped, low LT index links make good candidates from a survivability standpoint. If a link must be added, low ND index nodes would be good termination points for adding the link. The original selection process, which made the decisions based strictly on link use and link cost, is thus replaced with one oriented towards maintaining network survivability.

- Several different sets of weighting factors were applied to the design criteria to form an objective function for design optimization.

- Alternative networks were designed to meet the common traffic and delay requirements, using each set of weights.

  *High Survivability*–link use, LT index, and ND index
  *High Economy*–link use and link distance
  *Minimum Average Number of Hops*–link use and hop count
  *Balanced Performance*–considering all criteria

These procedures can be applied to the interconnection subsystem that handles information and control flows within the network of modules corresponding to a combat system. For given load and timing scenarios, analytical experiments can be used to demonstrate this approach to combat system control design. In addition, the *extreme point* solutions form templates that can be used to construct a figure of merit for use in evaluating the design alternatives.

## 6.7 SUMMARY

For use in conceptual design, a preliminary state space model can be derived from prose statements of the required operating capabilities. For each statement $\mathbf{P}_i$, suppose the variable $\mathbf{x}_i$ indicates the degree to which the corresponding requirement will be satisfied in the final system design. If the statements are constructed at the same level of abstraction and are unifunctional and implementation fee as described in Section 3.3, the resulting state space model can be used for a preliminary (qualitative) assessment of functional interactions within the combat system. If required operating capabilities are given in hierarchical form, the higher level statements presumably correspond to capabilities of primary importance. The associated variables $\{\mathbf{x}_i\}$ thus form a simplified or reduced-order model for a similar analysis. It is possible that such simplified models will be useful at the initial stages of a backbone control design effort.

The quality of decentralized control structures depends not only on stability and performance index, but also on the costs of communication, reliability, the costs of control equipment and computer programs, the value of information (or lack of information), and human factors. Overall, there is a very long way to go before a comprehensive theory emerges. However, a good deal of effort has been put into development of the underlying theory, and many valuable lessons have been learned.

## 7.0 CONTROL STRUCTURE DEFINITION

This section identifies various design strategies for engineering of combat system control structures applicable at the fifth step in conceptual design (concept definition and advance planning). Mathematical notation uses the format established in the preceding section.

### 7.1 INFORMATION PATTERNS

The question considered here is to find the structure of information flow from the available measurements to the acceptable manipulations that satisfy a complete set of control objectives. The former involves continuous monitoring of system operation by observation of performance-critical outputs. The latter concerns action to bring system outputs from unwanted to desired states in a tolerable fashion under the influence of disturbances impinging on the system. In particular, design rules used to relate measurements to control actions must be identified. Figure 11 describes relationships among hierarchical control methods with different information patterns and solution strategies.

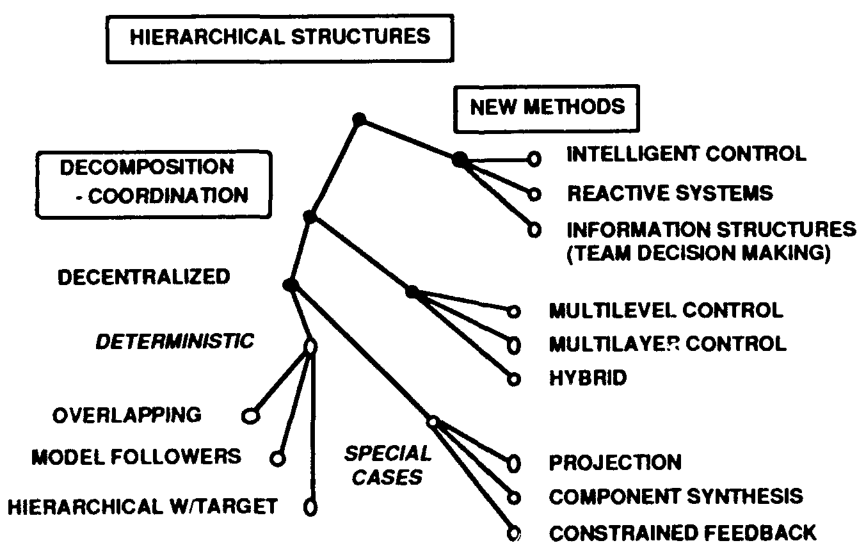



FIGURE 11. FEASIBLE CONTROL METHODS

The classical control theory mainly considers problems of centralized information pattern, in which one particular processor forces all others to adopt a single picture of system state. Though hierarchical methods break the overall problem into parts, information patterns and implementations are generally centralized. The key control objectives are then identification and control design.

The second approach involves a decentralized information pattern, in which a team of controllers is substituted for the central processor or decision maker. Each controller works with partial information, and coordination is achieved by message exchange rather than timing. The feedback channels are often structurally constrained so that each local controller is influenced only by the information available at the same station (or channel), and global information transfer among the channels may not be provided. Control of such distributed systems is based on methods for manipulating the state of each processor to achieve desired cooperative effects. This creates yet a third key control objective, namely to integrate the team of controllers into a consistent and coordinated control structure. Decentralized information patterns are found in many complex systems with geographically separated subsystems. Essentially, spatial decomposition of the overall system causes the information pattern to occur.

Decentralized patterns are produced by subsystems with disjoint information sets. Shared information patterns also occur and produce an overlapping system structure. For many classes of systems, sharing of information among the controllers is absolutely essential.

Alternatives can be generated using combinatorial algorithms that consider controllability and observability properties in an appropriate way. Once a number of feasible control structures have been generated, some can be eliminated on physical or operational grounds. Within the linear systems framework, further evaluation involves assessment of resilience and feasibility of decentralized control structures. Research on the behavior of linear time invariant models has led to definition of necessary and sufficient conditions as well as algorithmic procedures for assessment of controllability, observability, output controllability, and output functional controllability, in both the complete numerical sense and the more flexible structural sense. For nonlinear systems, however, the corresponding system theoretic properties are not completely understood.

### 7.1.1 Hierarchical Control

Many times, the global system of interest contains several important subsystems. If the subsystems are independent and interacting weakly, then multiple controllers can be ganged together to form a suitable control structure. When the subsystems have strong interactions, however, conflicts can arise between the controllers. Thus, a second level of control may be provided to resolve the conflicts by taking interactions into account. This gives rise to hierarchical structures with multiple control levels and objectives. As shown in Figure 12, hierarchical control systems have a pyramid structure so that on the first level there is a local control unit for each of the interconnected subsystems. In principle, the control units have different objectives, which may even be partially in conflict. Two notions that are fundamental in the design of hierarchical structures are task decomposition and coordination.

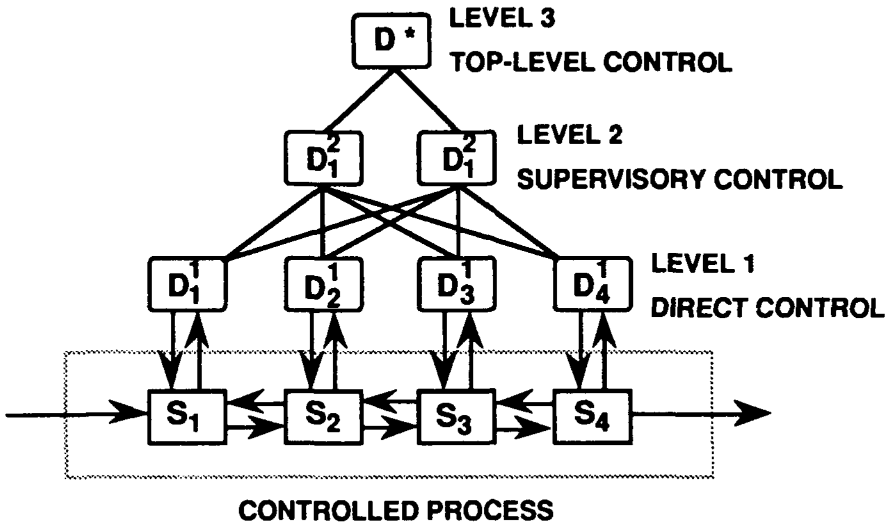



FIGURE 12. HIERARCHICAL CONTROL STRUCTURE

One of the strengths of hierarchical system theory is that human organizations often can be viewed as hierarchical decision making systems. The problems of coordination and control are strongly related to organizational structure and effectiveness. There are two classical (and fundamental) ideas in hierarchical control. First is the multilayer concept (Reference 28), where control is split into algorithms or layers, each of which acts on different timescales. Figure 13 shows an architecture for AAW in multilayer form.

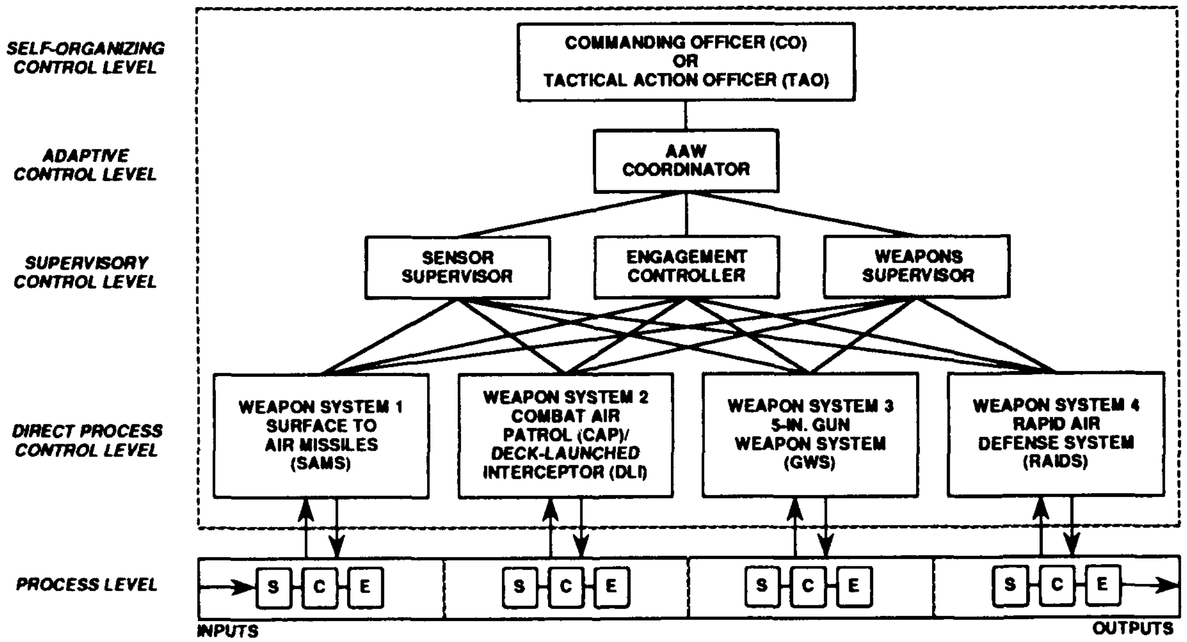



FIGURE 13. AAW ARCHITECTURE–MULTILAYER FORM

The second fundamental concept is the multilevel form (see Reference 29) where control of a complex, multiobjective system is divided into local goals, local control units are introduced, and their action is coordinated by an additional supremal unit. The control functions are thus distributed among two or more levels, and some controllers have only indirect access to information on the controlled process. These controllers depend on higher level units for information, while the latter act through the information flows to accomplish their control objectives. The information pattern is decentralized when each local unit has access to different information, and the local controls chosen depend on the specific information available (past and present). This approach was inspired by decomposition and coordination methods developed for mathematical programming.

7.1.2 Decentralized Control

Decentralized control is an important particular form of spatial decomposition that is used in many complex systems that are geographically distributed over long distances. A decentralized system with linear subsystems and linear interconnections can be modeled in the following manner:

$$\partial \mathbf{x}_i(t) \partial t = \mathbf{A}_i \mathbf{x}_i(t) + \mathbf{B}_i \mathbf{u}_i(t) + \sum \{1 \le j \le n, \quad i \ne j: \mathbf{A}_{ij} \mathbf{x}_j(t)\}$$

$$\mathbf{y}_i(t) = \mathbf{C}_i \mathbf{x}_i(t)$$

for i=1,...,n and matrixes $\mathbf{A}_i$, $\mathbf{B}_i$, and $\mathbf{C}_i$ with the proper dimensions. Interactions between subsystems are reflected in the off diagonal blocks $\mathbf{A}_{ij}$ of the overall system matrix **A**. This model is based on the assumption that each subsystem has a local input and that some linear transformation of the overall state vector is available for feedback. Sometimes only the subsystem state is available locally. Controller design is simplified for a number of special cases, as indicated in Figure 14 below.

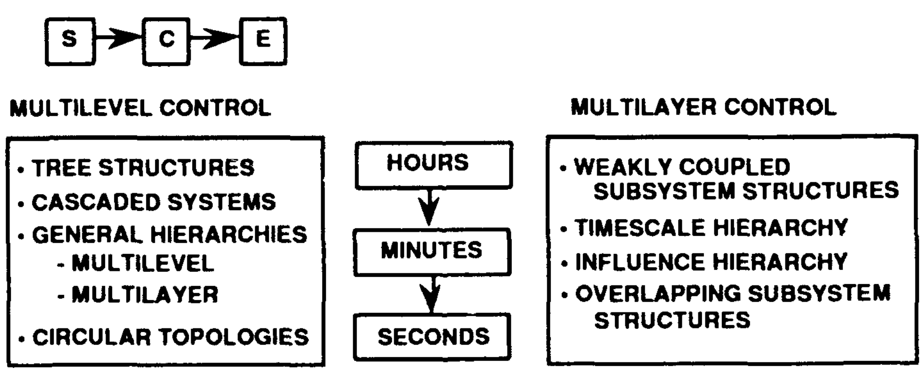



FIGURE 14. SIMPLIFYING TOPOLOGIES

System topology is reflected in a graph of its interconnections. Trees and cascades have unidirectional interconnections with no loops and permit a top-down control design. As shown by Reference 30, general series-parallel composite systems are controllable and/or observable for a wide range of interconnection structures, provided some mild conditions hold for the subsystems. Many large-scale transportation, water resources, and manufacturing systems can be treated in this manner. Although loops occur in general hierarchies, it is often possible to partition the problem so that the strongly connected subgraphs can be addressed separately. Simple loops form a circular structure that has been considered as a special case, and stability can be maintained in such configurations as long as the loop gains are small. Each of these cases gives rise to a system matrix with a special structure (e.g., block triangular) that can be exploited. In principle, every system can be broken down into a collection of simple parallel, cascade, and feedback interconnections.

Approximation methods give rise to another set of special cases. For example, if the system matrix **A** has block diagonal form, the system is disconnected and control can be achieved by individual subsystem controllers. Sometimes **A** is not block diagonal in form, but the terms outside the block diagonal region are relatively small. In such cases, the subsystems are said to be weakly coupled, and separate subsystem controllers may give satisfactory performance.

A large body of literature exists on systems with dynamics in which two or more timescales are present. Often these systems can be partitioned by timescale into independent, lower order problems. Singular perturbation is the corresponding analytical approach.

A number of design approaches for large, interconnected systems have been developed based on analysis of some measure of interaction effects between control loops of different subsystems. These approaches apply to a large class of practical systems.

Two different strategies are used to solve decentralized control problems, as indicated below in Figure 15. The first, decomposition, uses mathematical programming methods to break the main problem into subproblems for which consistent and optimal solutions can be derived. Space, time, function, geography, frequency, and hybrid structures are among the factors used to partition the problem. The alternative is a composition strategy, the main feature of which is prior recognition of a set of dynamic models, each referred to some component of the overall system and characterized by a particular control objective. Effective ways of coordinating multiple controllers and organizing components into a unified operating entity are then sought.

Hierarchical control principles apply to decentralized as well as centralized system control structures. This is illustrated below in two figures. Figure 16 corresponds to a decentralized information pattern and a nonhierarchical structure; Figure 17 corresponds to a decentralized information pattern and a multilevel system control structure. In the latter case, the primary coordinator thus receives feedback information from the secondary or local controllers. The absence of feedback is one of the distinctive features of systems with centralized information patterns. Hierarchical designs with centralized information patterns are intended mostly for open loop control structures, while hierarchical designs with decentralized information patterns generally involve feedback. Both forms are compatible with the hierarchical command structures used for military organizations.

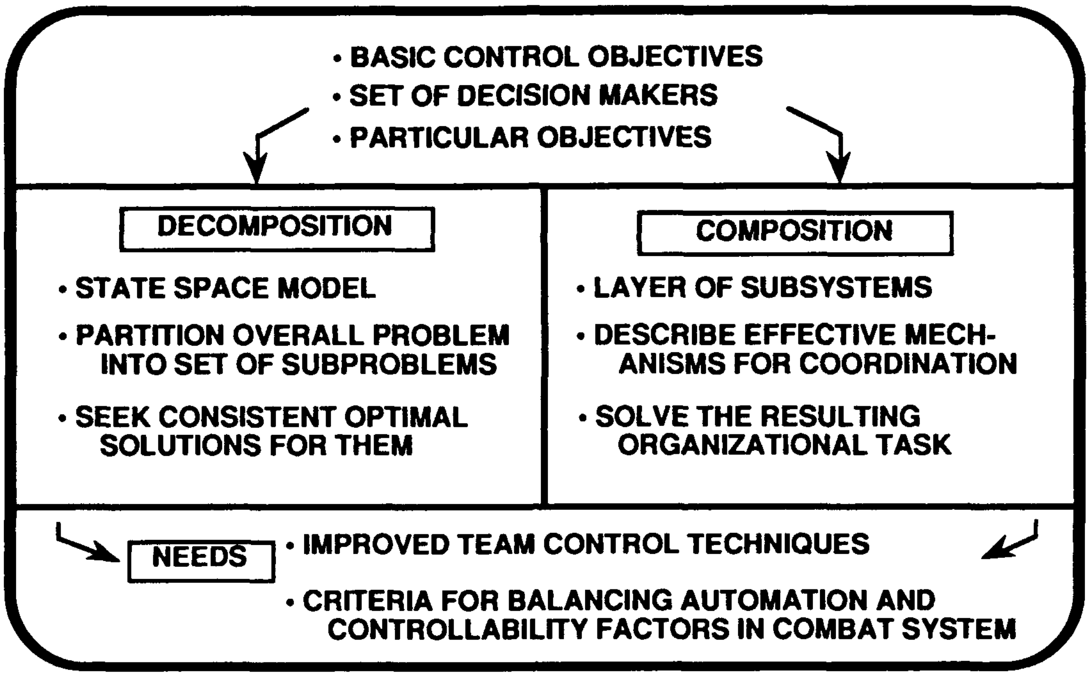
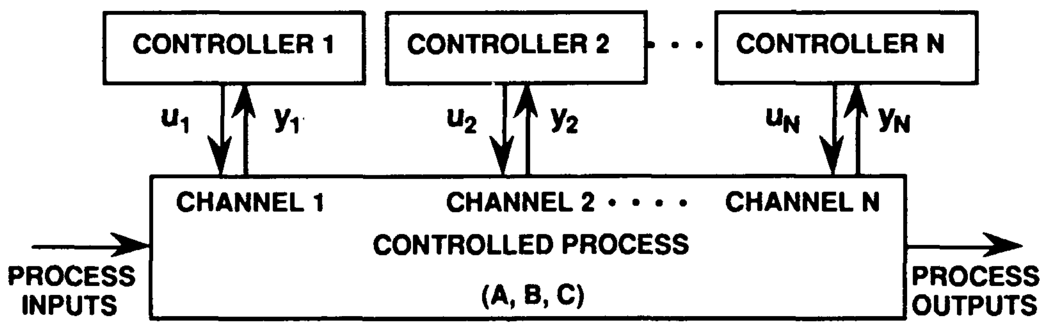



FIGURE 15. DECENTRALIZED CONTROL STRATEGIES




***SIMPLIFYING ASSUMPTIONS***

- **NO PRECEDENCE RELATIONSHIPS AMONG CONTROLLERS**
- **NO TIMING RELATIONSHIPS AMONG THE CONTROLLERS**
- **NO HIERARCHICAL INFORMATION / DECISION STRUCTURE**
- **PROCESS INPUTS AND OUTPUTS ARE IN CORRESPONDENCE**
- **u → CONTROL INPUT, y → PROCESS MEASUREMENT DATA**

FIGURE 16. DECENTRALIZED SINGLE-LEVEL CONTROL

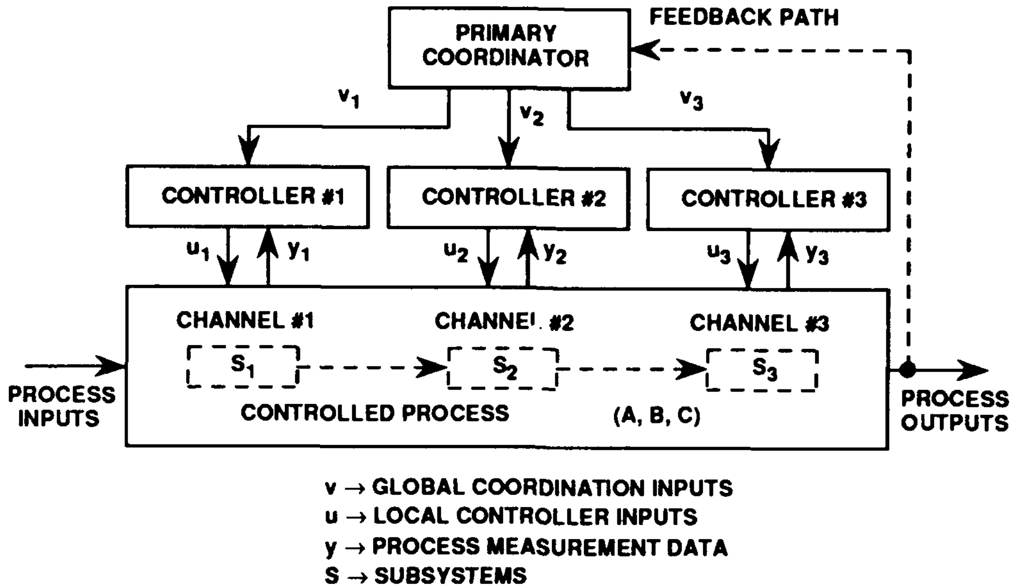



FIGURE 17. DECENTRALIZED MULTILEVEL CONTROL

7.1.3 Hierarchical Overlapping Coordination (HOC)

Most decentralized control schemes assume the information sets for subsystems are disjointed. There is, however, a large class of systems that involve overlapping information sets. In the process of modelling a large-scale and complex system, two or more mathematical models are likely to emerge, each of which may focus on a specific aspect of the system while still providing an acceptable representation for the overall system. In particular, this is common in hierarchical multilevel structures, where several approaches to decomposition may be both feasible and desirable. Overlapping subsystems occur in many economic models of international trade; they also exist in large electrical power systems.

Reference 31 articulates the idea of decomposing a system model into more than one decomposition, each responsive to a different aspect of the system and/or database, and with coordination through different couplings of the respective decompositions. The coordination ultimately leads to an overall optimum in single objective models, and to preferred Pareto-optimal solutions in multiobjective models.

In the area of water resources, for example, modellers using classical control methods were forced to choose between hydrologically based and politically based model decompositions, though each revealed important aspects not represented by the other. The Maumee River basin, the largest basin within North America's Great Lakes region, illustrates this point. It can be decomposed into five planning subareas on the basis of political and geographical factors, but can just as well be decomposed into eight watersheds on the basis of hydrological criteria. This suggests a decomposition like that shown in Figure 18 below. Each decomposition has its own merits and involves a very specific database that is collected and cared for by different agencies and constituencies. Fixing the interaction levels in either decomposition reduces the overall problem to independent uncoupled subproblems. Solution procedures are based on this property.

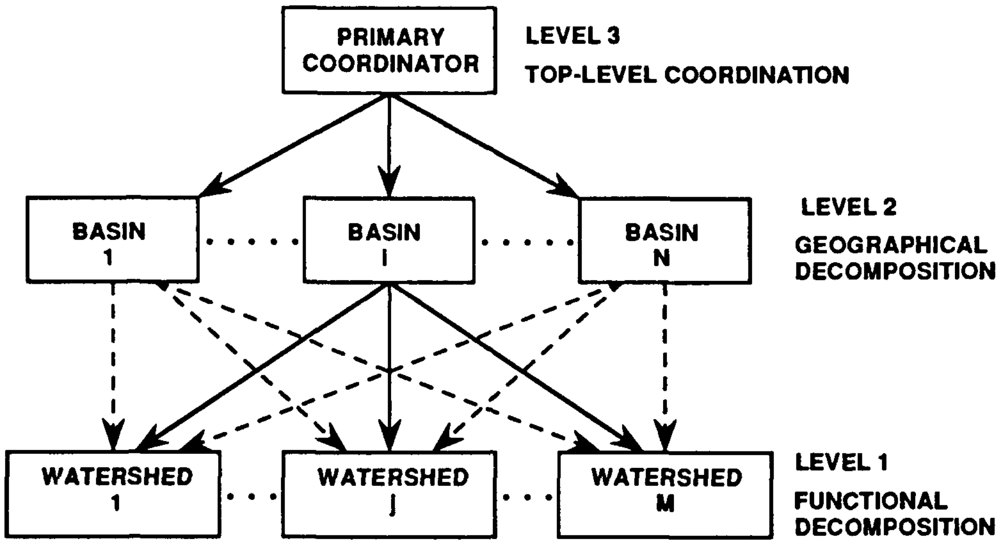



FIGURE 18. OVERLAPPING CONTROL OF WATER RESOURCES

Databases are a critical concern in the modelling of large-scale systems. Manipulation of databases to serve and satisfy demands and constraints (artificially imposed through the modelling process) may cause an eventual deterioration in model credibility. Hierarchical overlapping control enables the utmost use of these databases with minimum manipulation or misuse. This is achieved by using two or more simultaneous decompositions such as hydrological and political-geographical, each having a different number of subsystems. This approach allows subsystems to include their mutual interactions and to incorporate a core of state variables or elements common to all subsystems. For example, standby units can be placed in this common core. New approaches to robust control designs are possible because the overall system can be made dynamically reliable against failure of some (but not all) of its subsystems.

This has an obvious counterpart in combat systems, where decomposition by mission and function involves significant tradeoffs. A hierarchical overlapping structure for combat systems is shown in Figure 19. The functional decomposition shown is illustrative rather than prescriptive. It is presented only as an observation of recent trends (e.g., common launch systems, integrated communications, and common CIC workstations) as they relate to concepts of overlapping hierarchical control. In this context, specific interaction levels could be associated with each of the combat system operating modes that are identified in Figure 20.

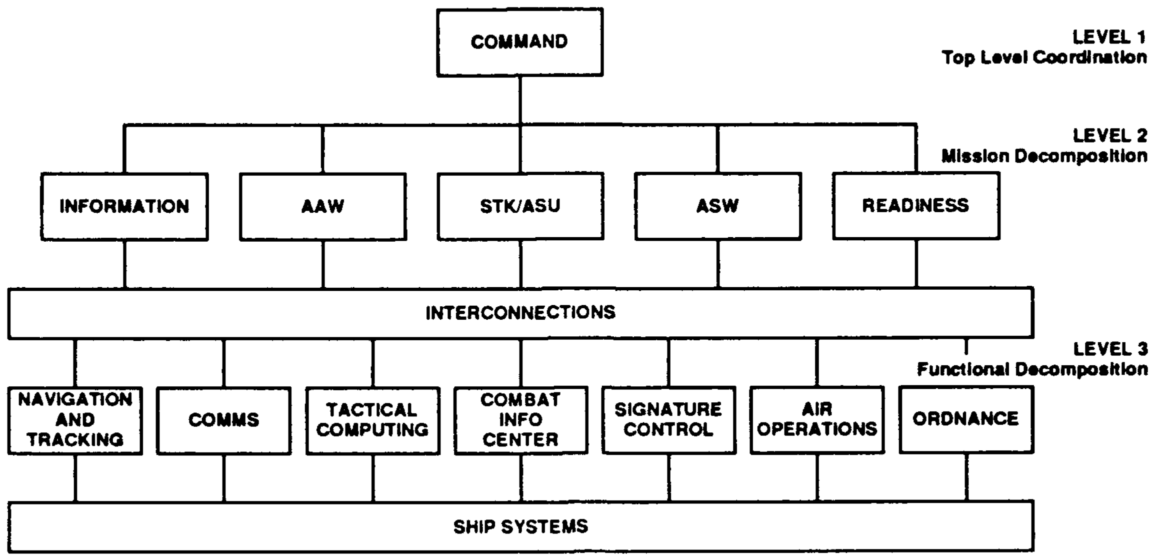



FIGURE 19. OVERLAPPING COMBAT SYSTEM CONTROL

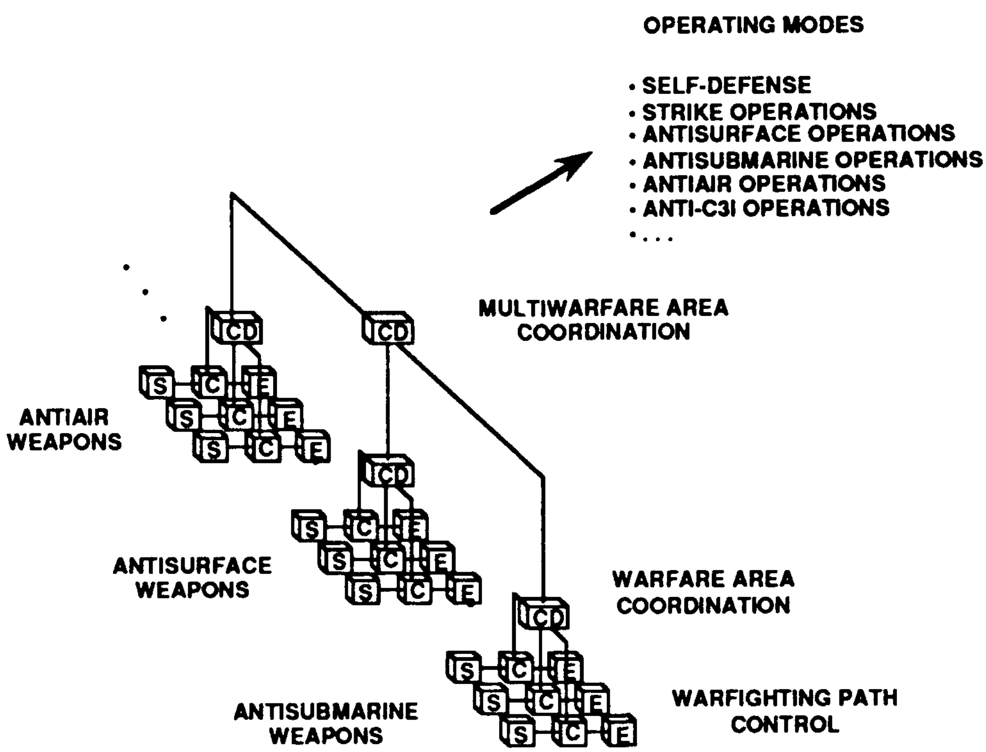



FIGURE 20. POSSIBLE OPERATING MODES

Recently, a systematic and unifying approach to this problem has been proposed. The underlying idea is to expand the state space of the original system so that the overlapping subsystems appear disjointed. A stabilizing decentralized control law can be formulated for the expanded system and then contracted for implementation in the original system. This approach can be used for systems that are not necessarily weakly coupled but can be treated as such if certain factors are repeated in the different subsystem models. The approach is used in freeway traffic-control applications and many contributions to the associated theory have been made by Reference 32 and more recently by Reference 33.

Reference 34 indicates that overlapping control methods also apply to large-scale systems that are composed of many similar subsystems, in each of which the available information contains one or more common aggregate signals. Local control of each subsystem is influenced by the common aggregate signals, and the responses jointly determine the new aggregate signals. Many economic phenomena may be modeled in this manner.

### 7.1.4 Autonomous Control

Reference 35 ranks control approaches by level of sophistication as shown in Table 2. Intelligent control techniques fall into the most sophisticated class and are used when the plant is so complex that it is inappropriate or impossible to describe it with conventional mathematical models. Automated design environments for control engineering may be the most promising area for such applications. Automatic control techniques involve an organized body of shareable knowledge; interactive computing methods can make it more accessible and so more useful. Initial applications may involve systems with a limited repertory; for example, neural nets offer promise as a way of producing a general purpose proportional integral (PI) controller.

TABLE 2. CONTROL STRUCTURES–BY LEVEL OF SOPHISTICATION

| PLANT CHARACTERISTICS | CONTROL TECHNIQUES |
|---|---|
| Simple, linear processes | Deterministic with feedback |
| Linear but more complex | Same plus state estimators |
| Linear, but with significant process noise | Same, plus Kalman filters |
| Processes to be completed with minimum time/energy | Optimal control theory |
| Quantitative stochastic processes or factors | Stochastic control theory |
| Large process parameter changes (operating modes) | Adaptive control methods |
| Highly complex (nonlinear stochastic and nonstationary) | Self-organizing or learning control |
| Large, hierarchical processes | Multilevel/multilayer |
| Unconventional modelling techniques required (e.g., artificial intelligence) | Intelligence control |

An issue agenda for intelligent control methods presented in Reference 36 includes the following items.

- A practical control theory must be invariant of system modelling. The conventional approach offers a model of a system, then treats the model as the source of theories for problem solving, instead of the reverse. Real problems often involve nested decision making and call for a nested hierarchy of control loops.

- Conventional research in control theory is driven by capabilities of the existing analytical and computational apparatus, rather than the real problems of users, which are often ill posed.

- Formulation of control objectives is not based as it should be on dialogue between users and control providers. It is important to provide for negotiation of cost functions between nested control loops.

- Planning is not considered to be a part of the control problem and is left to the user. In-the-loop planning must be provided.

- The practical difficulties of dealing with information are too often considered outside the scope of existing theory. Nested models of information acquisition, estimation, identification, representation, and control are needed.

- Novel control approaches based on artificial intelligence methods are not considered legitimate tools for control theory development.

7.1.5 Comparison

The hierarchical multilevel approach has been successful primarily in social systems and water resources systems. Reference 37 claimed five advantages for the multilevel structure:

- Decomposition of systems with fixed designs at one level and coordination at another is often the only alternative available.

- Systems are commonly described only on a stratified basis.

- Available decision units have limited capabilities, hence the problem is formulated in a multilayer hierarchy of subproblems.

- Overall system resources are better used through this structure.

- System reliability and flexibility will be improved.

Reference 16 reports that some disagreement exists among system and control specialists regarding these points. Reference 38, for example, has mentioned that the first three advantages are a matter of opinion, and there is no evidence justifying the other two.

## 7.2 SOLUTION SPACE

The character of the solution space depends on the scope of the problem to be addressed and the design strategy employed. The problem in its entirety is very large and complex. Some alternative design strategies are presented below.

### 7.2.1 Network of Modules

The starting point in a composition approach is the prior recognition of a set of dynamic models, each referred to a component of the large-scale system and characterized by a particular control objective. Reference 39 observes that if each component is controlled by a dedicated decision maker, then the pairing of a decision maker and an associated component model forms a module within the overall system. Clearly, local aggregations of modules can exist. In addition, a higher layer of decision makers can be associated with each aggregation. Thus a network of modules, arranged in layers to form a generalized hierarchical model, can be formed. This yields a modular design approach suitable for top-down design. However, it can also be used for bottom-up design by tailoring components to fit a top-down design template.

Generally, complex systems are said to be *modular* when they are composed of building blocks that can be added, removed, or interchanged to convert from one organization to another operable organization, with different but usually similar functional properties. These building blocks, often called *modules*, represent physical, logical, or functional units with known properties and considerable internal complexity. Using installations may choose the modules that best meet present needs, including or omitting any optional modules, and so tailor the system configuration to its own operational needs. Finally, a malfunctioning unit may replaced with an identical, operable unit, improving ease of repair.

With this approach, the combat system becomes a layered hierarchical network formed by nested composition of modules. Each module must be able to solve a decoupled control problem for its components, to include coordinating the behavior of any lower layer modules nested within it. This means information and control signals must be exchanged between each decision agent and all entities contained with the corresponding module. In particular, the system must provide as follows for each module:

- *Application:* Establish current operating objectives, configuration, and set points (control templates) for the module.

- *Presentation:* Maintain interfaces with decision agents providing situation assessment and control of module functions.

- *Network:* Exchange of information with related control elements to facilitate coordination between modules. Information and control signals must be exchanged between the decision agent and all component modules without excessive control efficiency loss in transport and disaggregation processes.

- *Protocols:* Provide for authentication, activation, and management of links to decision and action nodes.

- *Virtual Connectivity:* Provide physical and logical connectivity and route control for linkage to decision and action nodes.

- *Algorithm:* Exercise information processing tools to obtain needed judgements, forecasts, and/or perceptions. Suitably aggregated information must flow between the decision agent and any associated lower layers without excessive information losses.

- *New Information:* Task sensors and links (by invention and testing of hypotheses) to acquire the explicit knowledge needed for control tasks.

- *Prior Knowledge:* Provide for access to, and maintenance of, database elements containing prior tactical knowledge. In particular, the decision agent must be provided with the knowledge assets needed for control: plant models, control efficiency measures, and a way of selecting appropriate control actions.

A template for module architecture is shown in Figure 21 below. Content of the process and control models indicated in Figure 21 depend on the position of the module within the network. At the weapon system level, for example, the process of interest could provide for several alternative action paths.

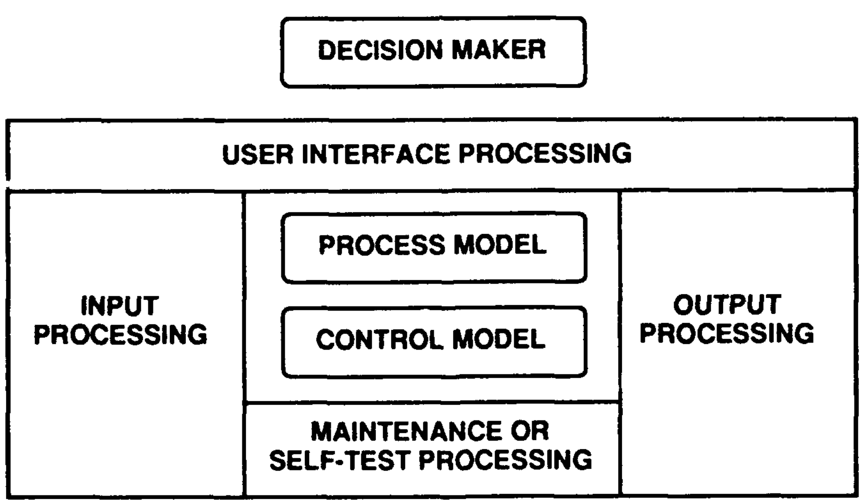



FIGURE 21. MODULE ARCHITECTURE TEMPLATE

The dynamical models and coordination efficiency measures for the modules are related by the following module inclusion principle: for a module at layer j, the corresponding dynamical model is the union of component models from lower layers of the network; and the coordination efficiency measure is the sum of the corresponding component efficiency measures and a measure unique to the current module. Given that links are defined by module inclusion according to this principle, the network will form a graph that contains no internal cycles or loops. The layers of the network are arranged in a generalized hierarchical structure, with each layer itself forming a network of modules. Figure 22 provides an example.

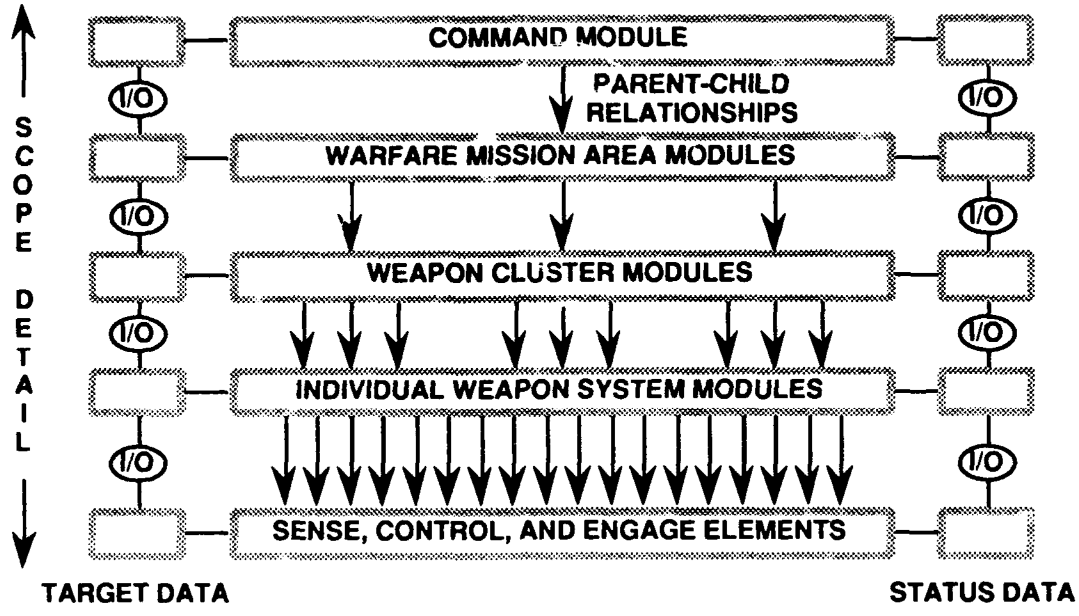



FIGURE 22. MULTILAYER, MODULAR APPROACH

In general, systems will have a layered hierarchy of at least three levels. Entities in each layer have distinct responsibilities and functions and are coupled only through message passing. Messages are exchanged only by well-defined interfaces between entities located on the same layer or adjacent layers. These messages imply functionality performed by a supplier entity on behalf of a using entity, and thus support identification of the functionality required for implementation. The system's physical assets, chiefly equipment, reside in the bottom layer. Fundamental operating tasks appear in the top layer, which is the layer seen by the user. Intermediate layers map the task objectives into physical reality, representing the successive resource manipulations needed to complete an operation. This creates an audit trail for design to ensure that implementation supports the user's operating concept for the system.

Architectures that meet these conditions are said to contain strict separation of concerns, both horizontally and vertically. Vertical separation of concerns implies separating task objectives into vertical layers from the abstract to the concrete. Horizontal separation of concerns means separating functional objectives into distinct, independent entities. Reference 40 indicates that architectures with strict separation of concerns are both easier to develop and easier to maintain. In fact, virtually all architectures contain some separation, although not to the degree envisioned here. Further, the increasing complexity of systems is expected to make such architectures necessary in future developments.

Multiple use entities can be formed in this process through vertical and/or horizontal consolidation of entities with common functionality. In design of the AEGIS Weapon System, for example, shipboard radar search and track functions have been consolidated across the detect, control, and engagement phases of air target processing.

The quality of system coordination achieved is limited by errors and inconsistencies among modules in terms of plant modelling, efficiency measures, communications, and control efficiency losses. The complexity of the plant model necessary for design depends on both the complexity of the physical system and on how demanding the design specifications are. An important tradeoff exists between complexity of a model and feasibility of exercising it to aid design.

7.2.2 Combat System Interconnections

Reference 41 considers a dynamic state variable model for a system S composed of N interactive subsystems. The main results given in Reference 41 are briefly reviewed here.

For each subsystem $S_i$, let $\mathbf{x}_i$ denote the local set of state variables, $\mathbf{u}_i$ the local set of controls, and $\mathbf{v}_i(\mathbf{z}_i)$ the corresponding I/O coupling vectors. If $\mathbf{x}_i$ is an element of the vector space $\mathbf{X}_i$, and $\mathbf{u}_i$ belongs to the vector space $\mathbf{U}_i$, then the state $\mathbf{x}$ of the composite system S is in the product space $\mathbf{X} = \mathbf{X}_1 \times \mathbf{X}_2 \times .... \times \mathbf{X}_N$. Similarly, the control $\mathbf{u}$ for S is an element of $\mathbf{U} = \mathbf{U}_1 \times \mathbf{U}_2 \times .... \times \mathbf{U}_N$, and the composite I/O vector $(\mathbf{v},\mathbf{z})$ belongs to the interaction space $\mathbf{V}$. The component model for the subsystem $S_i$ is then given by the following equations:

$$\partial \mathbf{x}_i/\partial t = f_i(\mathbf{x}_i, \mathbf{u}_i, \mathbf{v}_i) \quad (1)$$

$$\mathbf{z}_i = \mathbf{e}_i(\mathbf{x}_i, \mathbf{u}_i, \mathbf{v}_i) \quad (2)$$

$$\mathbf{v}_i = \mathbf{h}_i(\mathbf{z}_j, \mathbf{u}_j) \text{ for } j = 1,....,N \quad (3)$$

where functions $f_i$, $\mathbf{e}_i$, $\mathbf{h}_i$ together with their first and second order derivatives are continuous in all arguments. The model is stated in nonlinear form for maximum generality.

The results given by Reference 41 are of special interest due to the next step. By substituting Equation 2 into Equation 3, it is possible to obtain a composite equation

$$\mathbf{z} = \mathbf{g}(\mathbf{x}, \mathbf{u}, \mathbf{v}) \quad (4)$$

that gives a static interconnection system for S. Since the research was motivated by an application with dynamical interconnections, the model was revised to treat the interconnection system as a separate subsystem, making N+1 in all. The equations for the revised system model S' then become

$$\partial \mathbf{x}_i/\partial t = f_i(\mathbf{x}_i, \mathbf{u}_i, \mathbf{v}_i) \quad (5)$$

$$\mathbf{v}_i = \mathbf{h}_i(\mathbf{z}, \mathbf{u}') \quad (6)$$

$$\partial \mathbf{z}/\partial t = \mathbf{g}(\mathbf{x}, \mathbf{u}, \mathbf{u}', \mathbf{v}, \mathbf{w}) \quad (7)$$

$$\mathbf{w} = \mathbf{h}(\mathbf{z}, \mathbf{u}) \quad (8)$$

where $\mathbf{w}$ is the input coupling vector and $\mathbf{u}'$ the control for the new subsystem. The algorithm outlined above is easily adapted to the revised system model S' resulting from this change. The higher level of the algorithm then sets values for coordination parameters $\mathbf{w}$, $\mathbf{b}'$ and finds an optimal solution for the interconnection subsystem. The lower level sets values for coordination parameters $\mathbf{b}_i$, $\mathbf{v}_i$ and finds optimal solutions for the first N subsystems (in parallel).

A third model is given for cases in which the original subsystems are governed by different dynamics than the interconnection subsystem. This will hold for many practical systems in which the modes of the interconnection system are slow compared to those of the original N subsystems. A temporal decomposition of the overall system S' can then be achieved. The *fast* part consists of N independent subsystems, with separable performance index; and the *slow* part consists of the interconnection subsystem.

Reference 42 solves a linear version of the model given by Equations 5 to 8 using a quadratic index of performance

$$\mathbf{P_D} = \int_D \{\Sigma_I (\mathbf{x}_i \mathbf{Q}_i \mathbf{x}_i + \mathbf{u}_i \mathbf{R}_i \mathbf{u}_i)\}$$

where the domain of integration **D** is the interval $[t_0 \leq t \leq t_f]$ and the index set for summation is I={1,....,N+1}. Each matrix $\mathbf{Q}_i$ is assumed to be symmetric and positive semidefinite, while the block diagonal composite matrix $\mathbf{R}=[\mathbf{R}_1,....,\mathbf{R}_{N+1}]$ is symmetric and positive definite. To simplify the notation, $\mathbf{x}_{N+1}$ replaces **z** while $\mathbf{u}_{N+1}$ replaces **u'**. The linear-quadratic formulation leads to an efficient two-level solution algorithm. The task of the higher level is to choose approximate values for coupling inputs $\mathbf{v}_i$ and LaGrange parameters $\mathbf{b}_i$ associated with the coupling constraints, based on stationarity conditions for the problem. For given values of the $\mathbf{v}_i$ and $\mathbf{b}_i$, the LaGrangian function for the overall problem is separable into N independent minimization problems. The lower level thus functions to optimize the subsystems independently (in parallel). This algorithm can be solved iteratively and has given satisfactory results in a variety of examples. The equations are given in nonlinear form as the most general statement of the problem. For computation, a series of linear approximations, each correct over a small part of the problem space, would most likely be used. Linear or quadratic performance indexes for the overall system and each subsystem would also be used.

7.2.2.1 Unit Level Design. The starting point for control design is the unit level, at which the combat system is regarded as a composite of sensing, control, and engaging subsystems plus interconnections. The corresponding dynamical equations are as follows.

| | | |
|---|---|---|
| $\partial \mathbf{x}_1/\partial t = f_{i1}(\mathbf{x}_1, \mathbf{u}_1, \mathbf{v}_1)$ | [Sensing Subsystem] | (9a) |
| $\mathbf{v}_1 = \mathbf{h}_1(\mathbf{z}, \mathbf{u}')$ | [Sensor input coupling] | (9b) |
| $\partial \mathbf{x}_2/\partial t = f_{i2}(\mathbf{x}_2, \mathbf{u}_2, \mathbf{v}_2)$ | [Control Subsystem] | (10a) |
| $\mathbf{v}_2 = \mathbf{h}_2(\mathbf{z}, \mathbf{u}')$ | [Control input coupling] | (10b) |
| $\partial \mathbf{x}_3/\partial t = f_{i3}(\mathbf{x}_3, \mathbf{u}_3, \mathbf{v}_3)$ | [Engaging Subsystem] | (11a) |
| $\mathbf{v}_3 = \mathbf{h}_3(\mathbf{z}, \mathbf{u}')$ | [Engage input coupling] | (11b) |
| $\partial \mathbf{z}/\partial t = \mathbf{g}(\mathbf{x}, \mathbf{u}, \mathbf{u}', \mathbf{v}, \mathbf{w})$ | [Interconnection Subsystem] | (12a) |
| $\mathbf{w} = \mathbf{h}(\mathbf{z}, \mathbf{u})$ | [Interconnect input coupling] | (12b) |

where **w** is the input coupling vector and **u**' the control for the inter-connection subsystem. The equations are given in nonlinear form, according to Equations 5 through 8 above, as the most general statement of the problem. For computation, a series of linear approximations, each correct over a small region of the problem space, would most likely be used. Linear or quadratic efficiency measures corresponding to the overall system and each subsystem model would also be employed. The indexes have the form

$$\mathbf{P[D,I]} = \int_D \{\Sigma_I \, (\mathbf{x}_i\mathbf{Q}_i\mathbf{x}_i + \mathbf{u}_i\mathbf{R}_i\mathbf{u}_i)\} \qquad (13)$$

where I={1}, for example, gives the performance index for sensing. This begins a top-down partitioning of the combat system on functional lines, as opposed to a bottom-up procedure beginning with warfighting paths. Since combat systems are warfare systems, just as infantry battalion or tank companies are warfare systems, they can be broken down into sense, control, and engage elements. The functions assigned to each subsystem are indicated in Figure 23. This gives a starting point for control design by a top-down process. Comparing the control structure derived to the goalpost architecture identified previously may indicate if the solution is sensitive to the chosen point of entry or to the particular sequence of decisions considered.

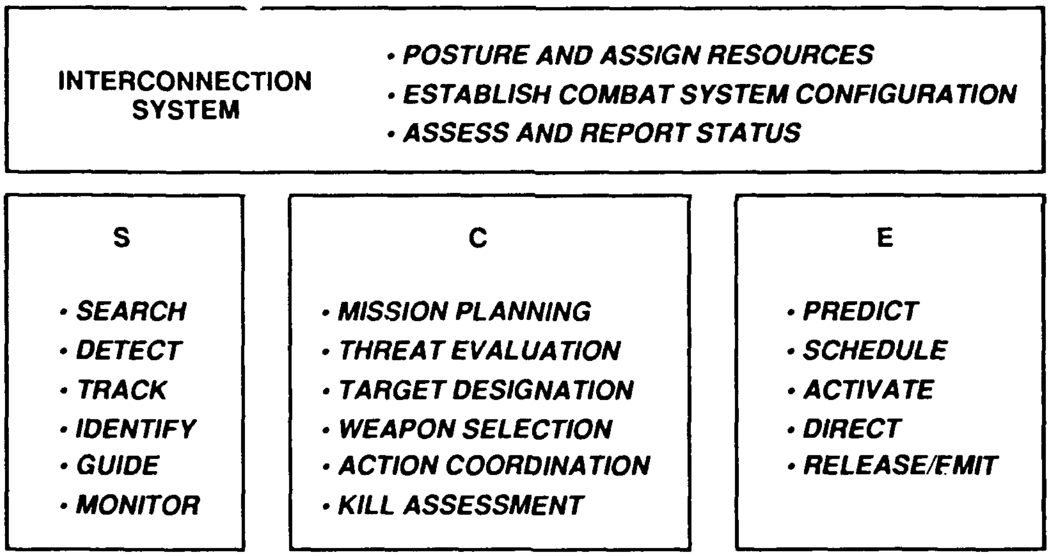



FIGURE 23. NETWORK OF MODULES–UNIT LEVEL

As stated in Section 7.2.1 above, each module contains a decision agent and must make provisions for exchange of information and coordination signal flows to and from component entities. This suggests a decomposition of interconnections at each level into information, readiness, and decision categories. Measures for consolidation of such assets at unit level involve a wide range of design issues and trades. The following are examples:

- The degree of reliance on organic sensors, the need for access to data from nonorganic sensors, and the capacity for integrated sensor management.

- The need to exchange information with external sources, including multiple media (visual, narrative, and voice), and the capacity for integrated management of communications resources.

- The need for comprehensive onboard database services, including the extraction, correlation, distribution, and display of tactical information. In this case, it appears difficult even to identify who should be made accountable for information coordination.

- A fully integrated CIC has yet to be designed, although CIC is the main center of combat system control activity.

- Overlapping hierarchical control methods may permit new forms of backup or reserve capability, such as display/console equipment on hot standby, and enhance system growth potential.

- With the spread of vertical launch systems, there is increasing potential for consolidating launcher and launch control assets.

In addition, control of the ship's service infrastructure (electrical power, communications, piping, and mobility) must be considered an important function at this level. Since it tends to cut across weapon system and warfare area product lines, the importance of service infrastructure is easy to overlook. A model formulated at the unit level encourages proper attention to these factors.

7.2.2.2 Warfare Area Design Level. Naval forces operate in three domains (air, surface, undersea) with radically different physical characteristics. This has long been a major factor in naval battle organization, and weapon systems usually are designed to work in a particular domain. Most surface combatants are equipped for operation against threats in each domain; that is, with subsystems specialized for AAW, antisubmarine warfare (ASW), and antisurface warfare (ASuW). Since the characteristics of action against land targets resemble those of ASuW, it is customary to treat strike warfare/antisurface warfare (STK/ASuW) as a single (multipurpose) subsystem.

The battle organization for surface combatants is based on the principle of decentralized command with warfare area coordinators delegated to handle each warfare area subsystem. The subsystems interact weakly and require minimal coordination, while the separate control systems permit simultaneous multiwarfare operations. The decentralized command concept reflects and supports the composite warfare commander concept typically employed for naval battleforces. The responsibilities allocated to the unit command authority can be summarized as follows:

- Compliance with force level decisions affecting ship operations.

- Delegating command authority to a lower organizational echelon for action within individual warfare mission areas.

- Exercising broad control over the individual warfare areas.

- Assigning multipurpose sensors/weapons to the appropriate warfare area.

- Resolving contention between the individual warfare areas for control over ship's sensors and weapons.
- Exercise of command override.

To determine when two decision makers are better than one is a question of fundamental importance to the organization for battle. The use of a warfare area coordinator is convenient for even the most basic coordination tasks. When path configuration control is added to the other burdens of command and control, the use of such auxiliary decision makers becomes a necessity. The responsibilities allocated to the warfare area coordinators are as follows:

- Accomplish the ship's mission within the assigned warfare area, including detailed conduct of engagements.
- Interface with command to conform to established rules of engagement.
- Control those sensors and weapons assigned by command.
- Communicate to command the status and planned employment of assigned sensors and weapon systems, providing summary information as needed to maintain a comprehensive tactical picture.
- Request from command, as appropriate, control over sensors and weapon systems not currently assigned.

Maintaining the view of combat systems as a layered network of modules, the breakdown into warfare areas is shown by Figure 24.

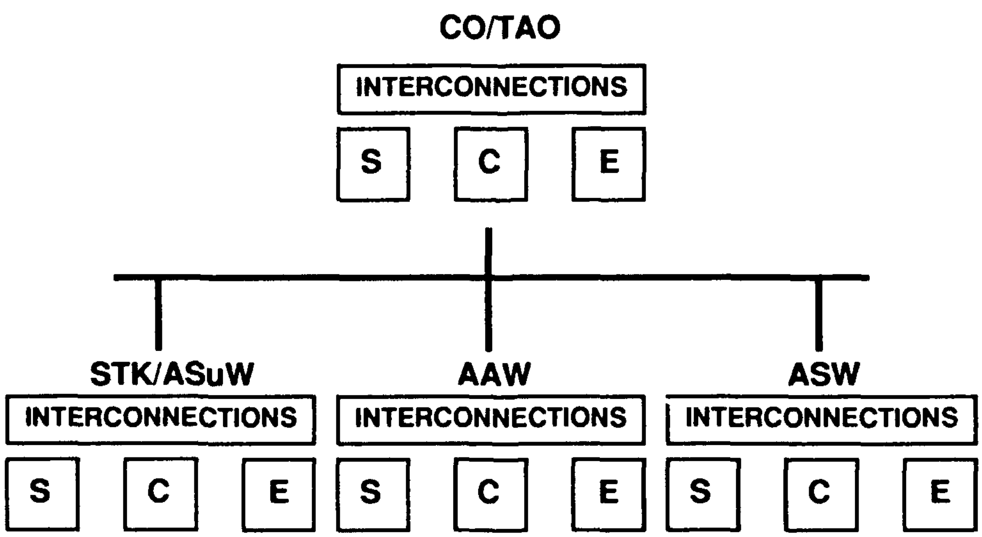



FIGURE 24. NETWORK OF MODULES–WARFARE AREA VIEW

Since each warfare area coordinator will have a local interconnection system to manage, as will the unit command authority, the backbone control system must be designed specifically to implement and support the desired battle organization. Both the unit commander and warfare area coordinators may need to retain some capability for autonomous action. The dynamical equations for the warfare area level are as follows:

| | | |
|---|---|---|
| $\partial x_{1j}/\partial t = f_{1j}(x_{1j}, u_{1j}, v_{1j})$ | [Sensing subsystems] | (14a) |
| $\mathbf{v}_{1j} = \mathbf{h}_{1j}(\mathbf{z}, \mathbf{u}')$ | [Sensor input couplings] | (14b) |
| $\partial x_{2j}/\partial t = f_{2j}(x_{2j}, u_{2j}, v_{2j})$ | [Control subsystems] | (15a) |
| $\mathbf{v}_{2j} = \mathbf{h}_{2j}(\mathbf{z}, \mathbf{u}')$ | [Control input couplings] | (15b) |
| $\partial x_{3j}/\partial t = f_{3j}(x_{3j}, u_{3j}, v_{3j})$ | [Engaging subsystems] | (16a) |
| $\mathbf{v}_{3j} = \mathbf{h}_{3j}(\mathbf{z}, \mathbf{u}')$ | [Engage input couplings] | (16b) |
| $\partial \mathbf{z}_j/\partial t = \mathbf{g}_j(\mathbf{x}, \mathbf{u}, \mathbf{u}', \mathbf{v}, \mathbf{w})$ | [Interconnection subsystems] | (17a) |
| $\mathbf{w}_j = \mathbf{h}_j(\mathbf{z}, \mathbf{u}),$ | [Interconnect input couplings] | (17b) |

where the subscript j denotes the particular module described by the set of equations given, with j=0 for the unit, j=1 for the STK/ASuW warfare mission area, j=2 for AAW, and j=3 for ASW. Thus the components shown for the unit level model are essentially broken down into warfare area components. Since the unit level module contains each of the warfare area modules, the equations indexed by j=0 simply represent components that do not fit into the warfare mission areas; e.g., reserve or multipurpose assets. If it were desired to explicitly represent other areas, such as mobility, a corresponding module would be added. Performance indexes take the form

$$\mathbf{P}[\mathbf{D}, \mathbf{I}\times\mathbf{J}] = \int_{\mathbf{D}} \{\Sigma_{\mathbf{I}\times\mathbf{J}} (\mathbf{x}_{ij}\mathbf{Q}_{ij}\mathbf{x}_{ij} + \mathbf{u}_{ij}\mathbf{R}_{ij}\mathbf{u}_{ij})\} \qquad (18)$$

making the same use of subscripts i and j as Equations 14 to 17.

7.2.2.3 Clustered Weapon Systems Level. Warfare area modules break down further into clusters of weapon systems, as shown for AAW in Figure 25. The figure refers to surface launched missiles, manned interceptors, and antileaker defenses. The last includes ESM equipment, electronic countermeasures (ECM), and a close-in weapon system (CIWS). The CIWS, however, could use SeaSparrow or Rolling Airframe Missile (RAM) instead of a Gatling gun system. The dynamical equations given by Equations 14 through 18 above apply to this level as well, once a subscript (say, k) is added to index the weapon clusters.

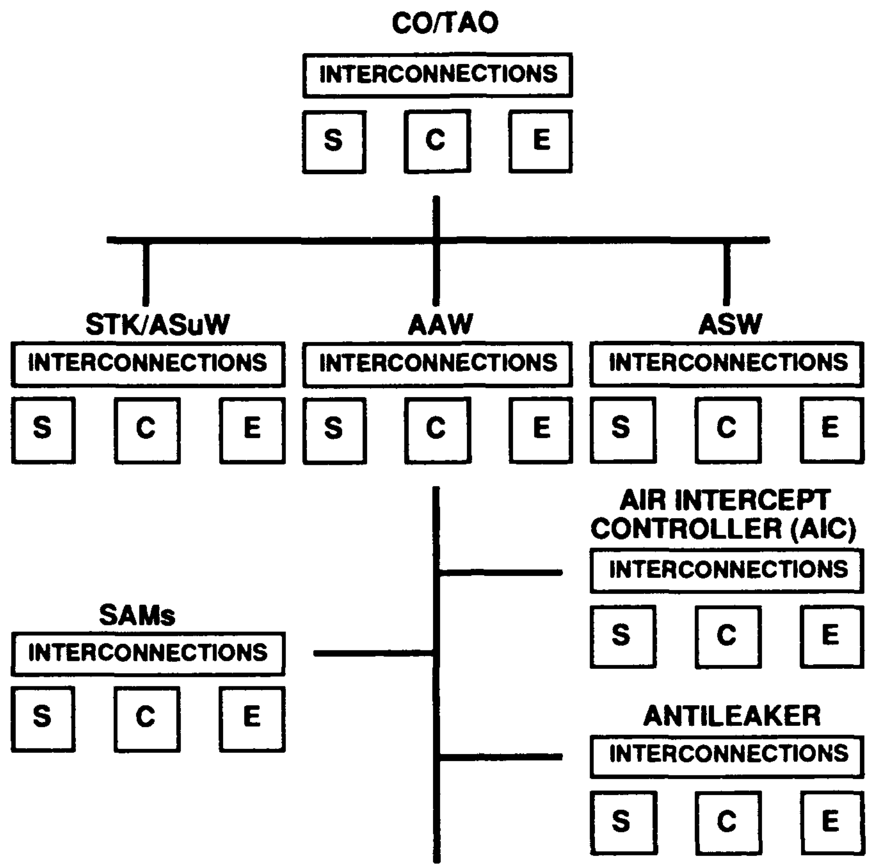



FIGURE 25. NETWORK OF MODULES–AAW WEAPON CLUSTERS

Antileaker defenses could mean a new level of system integration, a design issue of possible importance. The integration opportunity occurs below the level of the Antiair Warfare Coordinator (AAWC) but above the level of individual weapon systems.

Beyond this third layer lies another containing the individual components of the combat system. The modules involved in a discrete action sequence link together to form a warfighting path. This represents a virtual path rather than a physical path and may be significantly longer than the direct action path.

7.2.2.4 Interconnections: Evolving Product Line. Treating interconnection structure as a distinct subsystem brings the role of combat system integration into focus. The array of computing, communications, stored program, and knowledge-base assets that link command decision makers to sensors and actuators are an essential concern of combat system design. A layered architecture for systems of this type as shown in Figure 26 is widely used for both industry and military applications. This architecture has five separate yet interrelated layers, as shown by a pyramid with the command element at the top and the delivery system at the bottom. The first four levels are logically connected in top-down fashion. The fifth, the delivery system architecture, is the foundation architecture. Created to satisfy requirements of the other levels, its success is dependent on definition of relevant operating goals and objectives. Each level may contain multiple components as well as a set of discretionary and nondiscretionary standards for the enterprise.

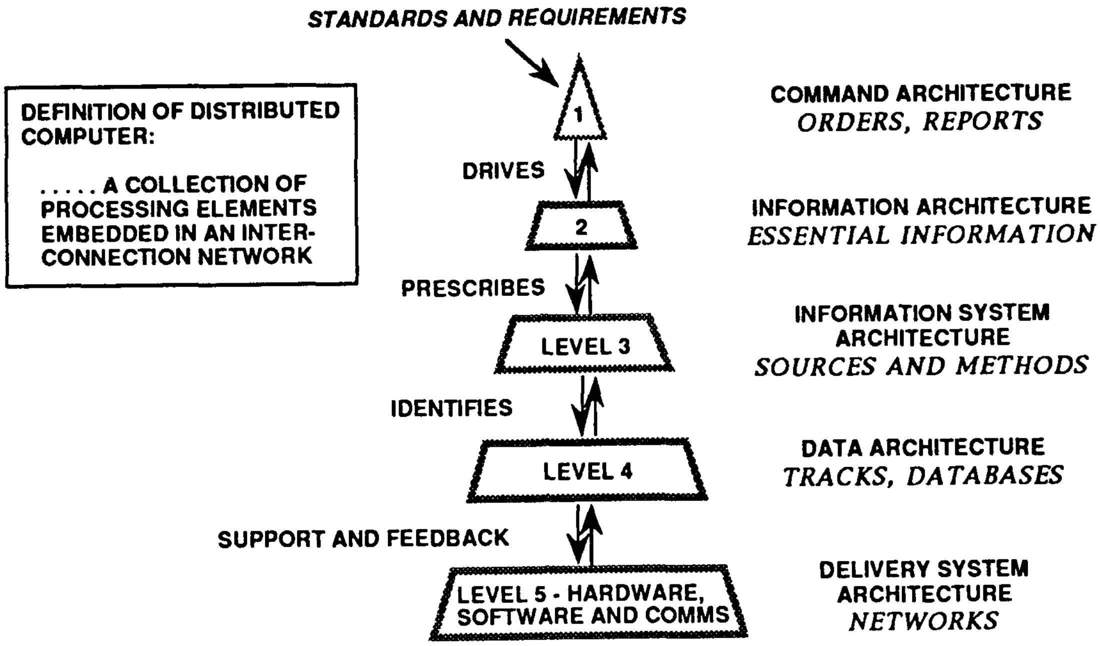



FIGURE 26. COMBAT SYSTEM INTERCONNECTIONS

Architecture at the *Command* level establishes a framework for satisfying both the internal information needs and those imposed by external activities. The latter include friendly forces, adversaries, and higher level command elements.

Architecture at the *Information* level establishes a framework to meet the information needs of the *Command* level. It specifies content, presentation form, and format of the information, thus establishing requirements for the *Information System* architecture.

Architecture at the *Information System* level establishes a framework for meeting the specific requirements of the *Information* level. Its components include automated and procedure-oriented information systems supporting internal and external information flows. They are used first to acquire and process data, then to produce and distribute information in accordance with requirements and standards. Logical database designs occur at this level.

Architecture at the *Data* level establishes a framework for maintenance, access, and use of the data of the enterprise. The data should meet the standards of all higher levels of the architecture, especially *Command.* Key components include data models that support physical database design; database and file structures; and data definitions, dictionaries, and data elements that underly the information systems of the enterprise. The creation of a data dictionary and associated naming conventions is an important aspect of the data architecture, because these conventions establish the vocabulary necessary for human communication.

The *Delivery System* architecture is a technical implementation to meet the requirements of all higher levels. Key components include computers, communications, and computer programs needed to support the *Data* and *Information System* levels of the enterprise architecture. Infrastructure and facility support assets needed to properly accommodate and connect the components in an integrated manner are also included. Since a collection of processing elements embedded in an interconnection network comprises a distributed computing system, distributed control computing is a critical technology associated with interconnection systems.

### 7.2.3 Real-Time Control Paradigm

A system design is a conceptual model detailing the relationship among component entities. Hierarchical structuring based on task decomposition is a proven approach for showing these relationships. The traditional hierarchical decomposition is not sufficient, however, to define a distributed real-time control system. A structured hierarchy only shows the system partitioning into modules, module arrangement, and interfaces. Essential system and/or component timing constraints and decision structure are ignored. Distributed computing involves the use of concurrent hierarchical control structures to satisfy these constraints.

Reference 43 attacks this problem by partitioning levels of the hierarchy into individual control loops with fixed cycle rates and feedback. At each level in the hierarchy, an interface exists where adjoining levels exchange I/O as commands and status, much like a feedback control loop. The fixed cycle rate implies a periodic sampling of commands and status to ensure that global control flow is goal directed. Unlike purely sequential hierarchical decompositions, each lower level is not completely dependent on the adjoining upper level. Although levels may share situation data, they can be considered to run independently, responding to a command and supplying a status much like a plant in a normal feedback control system.

The concept of a concurrent hierarchy is based on the method of task decomposition described above plus the use of parallel computing at each level of the hierarchy. The system is thus structured into layers of virtual control loops. When executing, each virtual control layer in the hierarchy can be considered as part of a long chain defining the hierarchical state, yet each level's action is based on its own control flow. Much as a computer executes an instruction within a given duty cycle, the virtual control loops correspond to layers of software modeled by analogy to a physical machine. The virtual control loop software exhibits cyclic feedback behavior that samples inputs including command and status and guarantees some output within a given time. The need for a fixed cycle response time leads to further partitioning and functional specialization within the virtual control loops.

Reference 44 considers ways to configure advanced real-time controllers from commercially available components. Controller throughput and computation speed are enhanced by using two or more dedicated computer subsystems to separate time-critical events from non-time-critical events. This provides a temporal decomposition strategy for use in design of real-time controllers.

For example, suppose that separate executive and real-time subsystems are formed. The executive subsystem is formed of a general purpose (host) computer, some printer/plotter units, disk drives, and computer terminals. The executive subsystem operates independently by virtue of its multitasking capabilities, yet works through an interface to maintain control of the real-time subsystem.

The real-time subsystem might include an array processor and analog to digital (A/D) and digital to analog (D/A) converters. It will operate independently, though under control of the executive subsystem, to perform all of the real-time control functions. This approach can greatly increase system throughput as well as the computation rate. Because data acquisition and actuation devices can be directly linked to the array processor, data transfer can be accomplished at speeds much higher than those permitted by a typical host computer.

### 7.3 SUMMARY

Strategies considered include decomposition of the overall combat system into a multilayer network of modules, decomposition of control functions into an executive operating system and a real-time operating system, and decomposition of functional or physical elements into subsystems that include a dynamic interconnection system. Formal analysis methods may be useful at this stage to ensure that any preferred design solution that may be synthesized forms a proper functional entity and meets basic operating requirements.

## 8.0 INTERCONNECTION TECHNOLOGY

Technology opportunities are sometimes considered in an extension of the conceptual design phase. It is particularly important at early stages of the system lifecycle to identify technologies that may be useful in resolving key design-related problems.

### 8.1 WORKSTATIONS

For surface combatants, the CIC is of central importance in tactical command and control. Workstations couple the command team to the interconnection system and, with the development of distributed computing technology, can supply a good deal of tactical computing capability as well. Even in the most modern combat systems, CICs today are only partially integrated. Future combat systems will provide for a fully integrated CIC that can be reconfigured to match changing mission needs. Evolution of a common workstation design, with standard man/machine interface characteristics, is likely. Each workstation will be configured for its intended use by the command of the user (resource coordinator).

### 8.2 EMBEDDED COMPUTING ALTERNATIVES

Since the 1950s, combat system functional performance has evolved from a bottom-up design approach. During this period, computers and memory (reckoned in megabits of throughput or capacity) have become much less expensive, but the cost of computer programs has gone up. Federated systems are difficult to change; system integration involves extensive computer programming effort because it is achieved by computer to computer interface. Substantial time and testing is also needed to certify that changes have not affected the existing computer programs.

The computer industry is rapidly moving away from large centralized or federated computer systems. The trend is to smaller, but very capable mini- and micro-computers connected together in a network. The term *distributed system* is used to refer to a computing system that has the following characteristics:

- It includes an arbitrary number of system and user processes.
- System architecture is modular, consisting of a possible varying number of processing elements. Spatial distribution is not essential to the concept.
- Communication is achieved by message passing on a shared interconnection structure (excluding shared memory).
- Some system-wide control is performed to provide for dynamic interprocess cooperation and runtime management.
- Interprocess message transit delays are variable, and some nonzero time always exists between production of an event by a process and materialization of this production at the destination process (different from observation of the event by the destination process).

These characteristics may be viewed as general rules observed in the design of distributed systems. Distributed systems are intrinsically more complex than centralized systems. Structured design methods can reduce complexity by factoring designs into functional layers and are, thus, more important than formerly. Industry is investing millions of dollars into this technology, much more than defense, and the Navy can directly benefit from this investment.

In the recent NFR-90 program, a distributed architecture was the agreed choice of all eight participating navies and had the complete backing of the industries involved. This concept was accepted as the way of the future in military systems. The nations specified this design concept in the NATO Staff Requirements. A distributed architecture concept was also attractive to the U.S. Navy. Spreading the cost and risk over eight nations would compensate to some degree for the impact of a changeover on the Navy's large investment in standard computers, displays, test sites, and programming centers.

Detailed engineering tradeoff studies were performed to identify an acceptable interconnection structure. System loading parameters were defined, and detailed computer simulation studies were performed. The conclusion was that only two available databus standards could meet the NFR-90 requirements. They were the SAE AS4074.2 High Speed Ring Bus (being developed to full military specifications for real-time systems) and the Safenet II implementation of ANSI-FDDI, a commercial standard. No final selection was made before the project ended, pending refinement of system data loads and timing requirements. However, Reference 45 reports serious technical reservations that the ANSI-FDDI-Safenet II standard could meet NFR-90 requirements.

The Vision Architecture is a conceptual structure based on end-use considerations rather than implementation technology. Figure 27 shows a variety of embedded computing configurations that can be used for its implementation. The structures shown are mostly illustrative; no doubt others can be identified. They serve to illustrate differences in spatial distribution of computing resources, information pattern, interconnection structure, and task partitioning–i.e., oriented on objects, missions, or functions. The differences are considered at more length below.

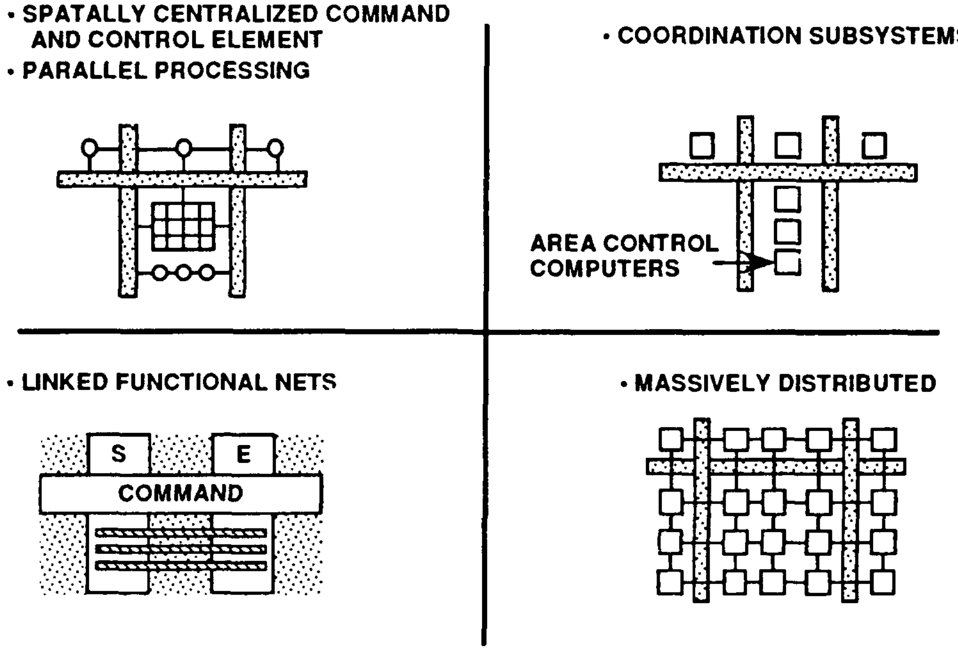



FIGURE 27. DECENTRALIZED CONTROL ALTERNATIVES
(COMBAT SYSTEM BACKBONE DESIGN)

8.2.1 Parallel Processing

Figure 27 indicates a central parallel processor with smart terminals for command support and the coordinator positions (sensing, readiness, and warfare mission areas). Although spatially concentrated, individual processing units in this configuration could be allocated in a flexible manner to specific processing tasks. Any information pattern, (centralized, decentralized, and overlapping) can be accommodated. Shared memory concepts, described below, provide the necessary connectivity.

Research conducted for the National Institute of Standards and Technology considered concepts for hierarchical control of large-scale systems using microcomputer networks. Reference 46 describes this work. The network operates in 20-msec timeslices. At the beginning of each slice, each logical module reads its input data from designated locations in a common memory system. The modules then compute their output data and write back into the common memory before the 20-msec cycle ends. A resent-synch pulse signals the beginning and end of each computation cycle. Each logical module is a state machine that samples its input for command and feedback variables, performs computations, writes its output into the database, and waits for the next computation cycle. None of the modules admit interrupts, so the network has a simple modular structure that enormously simplifies the writing and debugging of software.

The architecture has four layers: plant control, cell control, workstations, and smart devices. Figure 28 illustrates application of this design concept to combat systems. Orders are entered at the top and control flows from top to bottom throughout the plant. I/O devices within each layer provide access to two separate databases, one providing track data (tactical picture), the other readiness data. Entries and queries to or from the management information and control database enable command to control the plant by setting priorities or optimizing various parameters. One part of this database identifies the loading and use for each element of the plant. One set of computers (feedback processors) operates to extract from each layer the information needed at the next higher level. The memory is partitioned so each layer is able to access information at an appropriate level of detail.

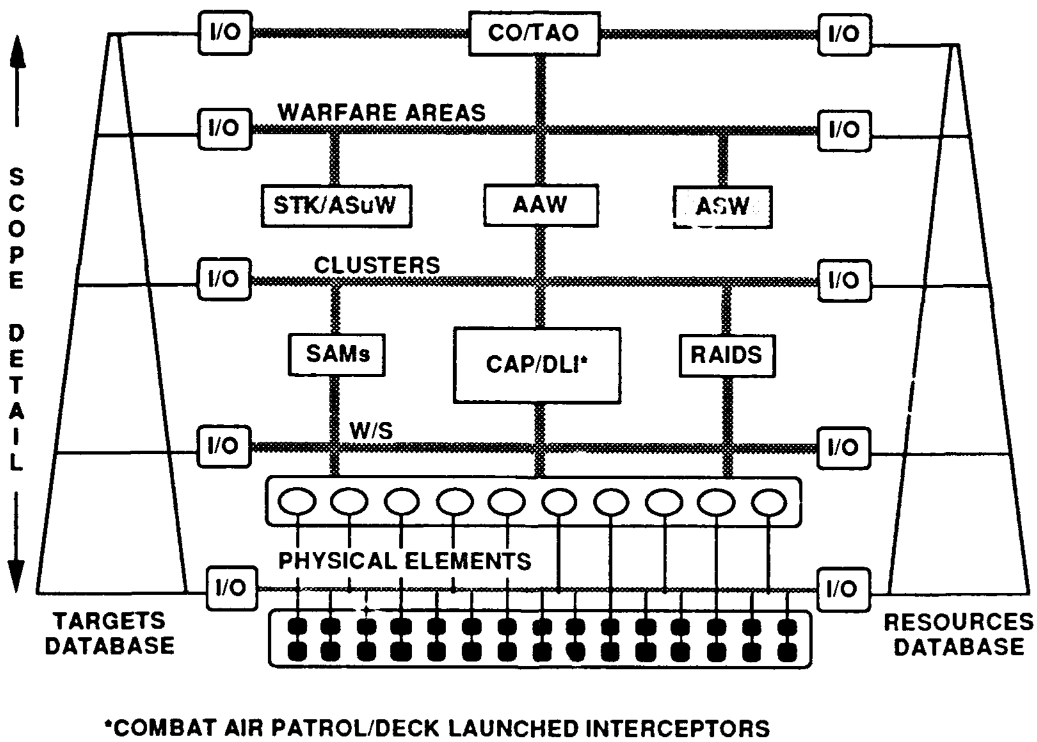



FIGURE 28. MULTILAYER, MODULAR STRUCTURE

In all cases, communication occurs through the medium of a common memory. The *mail drop* approach used has a disadvantage in that two data transfers are required to get information from one module to another. This is offset by the following advantages:

1. There are no communication protocols between computing modules, which communicate only through the common memory.

2. Adding each new state variable requires only a definition of where it is to be located in common memory so the module that generates it knows where to write it, and the modules that read it know where to look. Thus, new microcomputers can easily be added, logical modules can be shifted from one microcomputer to another, new functions such as safety watchdogs, and even new sensors can be included with limited effect on the rest of the system. As long as the system bus has surplus capacity, the physical structure of the system can be reconfigured with few changes in the software resident in the logical modules.

3. The common memory always contains a readily accessible map of the current state of the system. This makes it easy for a system monitor to trace the history of the state variables, to set break points, and to reason backwards to the source of program errors or faulty logic. This is extremely important in a sophisticated, real-time, sensory, interactive system where many processes are going on in parallel at many different hierarchical levels.

Although processing in this system is completely modular, memory and access characteristics may become bottlenecked. A great deal of addtional work would be needed, therefore, to specify the data access, control software, and memory features. Whether the final result is a hierarchy implemented on a network of small computers or one implemented on a single large computer is not important. What is important is that the control problem can be decomposed into subproblems that can be solved in a network of computing modules wherein each module has a clearly defined interface of I/O variables and a clearly defined functional relationship between the input and output.

### 8.2.2 Decentralized Hierarchical Processing

This alternative calls for dedicated computers in each subsystem (command support plus coordinating positions for sensing, readiness, and warfare mission areas). The computers are interconnected, and goal coordination is provided by a designated unit, most likely the command support computer. The subsystem computers are also interconnected to smaller processors embedded within the combat system plant. This corresponds to an hierarchical form of distributed computing, and provides a computer architecture matched to the control structure prevalent in existing surface combatants. A corollary property is that a bottom-up transition to the use of LANs can be supported. In addition, warfare mision aread could transition to distributed computing technology at different paces. These features are exploited in an evolutionary approach to distributed combat system architectures presented by Reference 47 and illustrated in Figure 29.

Two basic methods are available for interconnecting the components of a distributed information processing and control system: first, point-to-point dedicated connections and second, LANs, which are high-speed communication channels for connecting a variey of devices at distances from tens of meters to kilometers. This approach introduces LANs to the combat system, although some point-to-point connections could be retained.

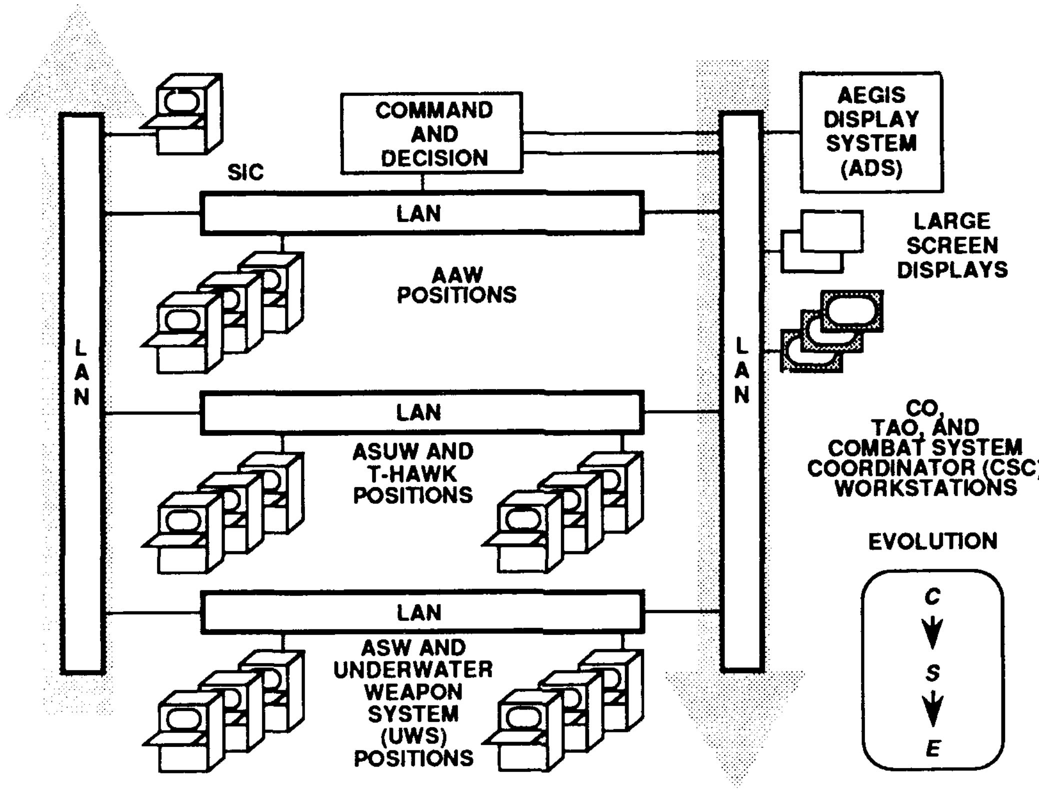



FIGURE 29. DISTRIBUTED CONTROL ARCHITECTURE FORMATION (COMPOSITE COMBAT SYSTEM PLANT AND OFFLOADING VIA LAN)

The major advantages of digital bus networks over conventional point-to-point dedicated connections include the following:

- Reduced wiring, space, and power requirements
- Flexibility of operations
- Evolutionary design process
- Improved maintenance, diagnostics, and monitoring

Some major technical barriers block the way to realization of totally integrated systems. The most pervasive issues concern standards for database structures and communication architectures. Although suitable standards will eventually emerge, advanced information systems are needed for use now. The problem is to meet present needs without compromising evolution toward practical standards. For a number of reasons, the private sector is turning to broadband LANs to solve this problem. The reasons are as follows:

- Multiple discrete nets can coexist on the same cable
- They are resistant to dirt and noise
- Standards are developing quickly, making it possible to integrate sensors and control systems from different suppliers

With the advancing capabilities of digital computers, even combat systems with federated architectures now contain many individual microprocessors supporting many different operational tasks. Making these computers communicate is an essential step toward new levels of combat system integration. LANs are the emerging technological products that best meet this need. They allow movement toward integration of diverse computer and communication networks into one large system with corresponding economies of scale.

Overall performance and openness to modification or expansion are also improved. However, the performance (e.g., data latency) of LANs can vary with the intensity and distribution of their traffic loads. The effects of data latency on the dynamic performance and stability of feedback control systems are often ignored in relatively slow processes such as those of many chemical plants. However, as the number of users on the network increases, the augmented traffic causes a larger data latency to a point where its impact on performance of some of the control loops (sharing the network) can no longer be ignored. The detrimental effects become evident quickly in very fast processes, such as the flight dynamics in tactical aircraft or missiles. Any lack of synchronization among system components, or loss of messages due to noise or saturation in the LAN, aggravates the detrimental effect.

Some researchers (see Reference 48) maintain that introduction of LANs alters the foundations of large-scale system control theory. The usual approach has been to assume that the information structure must be tailored on the physical structure of the plant. This no longer holds, and it becomes necessary to consider instead how to select the best information structure for given control tasks. Figure 30 illustrates this change in perspective.

Compared to the two decades that digital control had to wait before becoming an industrial standard, acceptance of the LAN concept was surprisingly quick. Since its origins in the data processing industry involved no safety hazards, adapting the LAN to process control applications was not a trivial step. Once established in the refinery industry, however, it was immediately accepted into general practice. The reason for its enthusiastic reception is simple–given all the processors in place, a medium for integration became necessary to make them work. LANs answer a need for easy connection between distant, heterogeneous units.

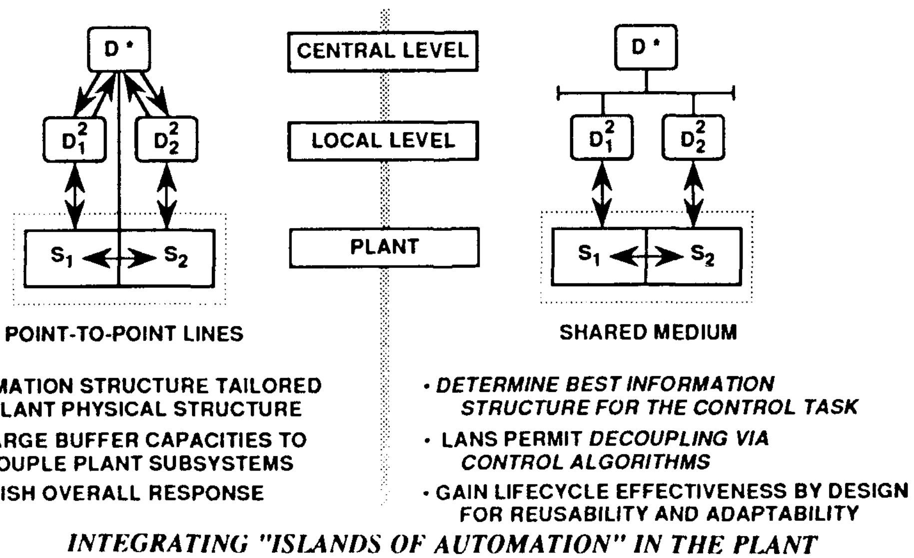



FIGURE 30. OPEN ARCHITECTURE VIA LANs

8.2.3 Linked Functional Networks

This alternative provides a functional structure for embedded computing in combat systems. All action paths are considered part of the command subsystem, from the I/O ports of the associated sensor elements to the I/O ports of associated engaging elements. Thus, control of the action paths is an essential function of the command subsystem. Connections between sensing elements, along with communication and fusion activities, belong to the sensing subsystem. Connections between engaging elements likewise belong to the engaging subsystem. This alternative encourages eventual formation of a fully integrated CIC, a fully integrated sensory suite (including offboard sensor reports), and a fully integrated ordnance facility. Each subsystem is considered to contain a separate interconnection network. However, the networks may be linked at various points. In addition, each of the separate nets may be configured with a high capacity control computer. That is, each of the functionally separated interconnection networks may be hierarchical in structure.

8.2.4 Massively Distributed Processing

This alternative has much in common with the DARPA High Performance Distributed Processing (HIPER-D) approach to distributed computing. The concept involves much spatial distribution of computing resources. Research on the concept of virtual processing illustrates this point.

Future computing systems could contain a mix of functionally specialized processing resources, each providing a specific class of services. Today's microprocessors are typically developed in family groupings, so upgrading to a new and more powerful microprocessor with little or no impact on computer programs is conceivable. For example, embedding single card computers into existing equipment is an approach used successfully in industry for several years.

In the virtual processing approach, as shown by Figure 31, there would no longer be a single, shared data storage facility serving all processors. Instead, storage is dedicated to individual processors or to clusters of processors. Such a modular computing facility could employ widely varying sets of funtional processors. Thus different processing engines would proovide communications, correlation, display, system and security control, database, database search, pattern recognition, knowledge base, and maintenance services. Within limits, processor orientations could be governed by mircorcode, alterable by a supervisory element to meet changing workloads. Increased levels of fault tolerance can be obtained with proportionate increases in cost.

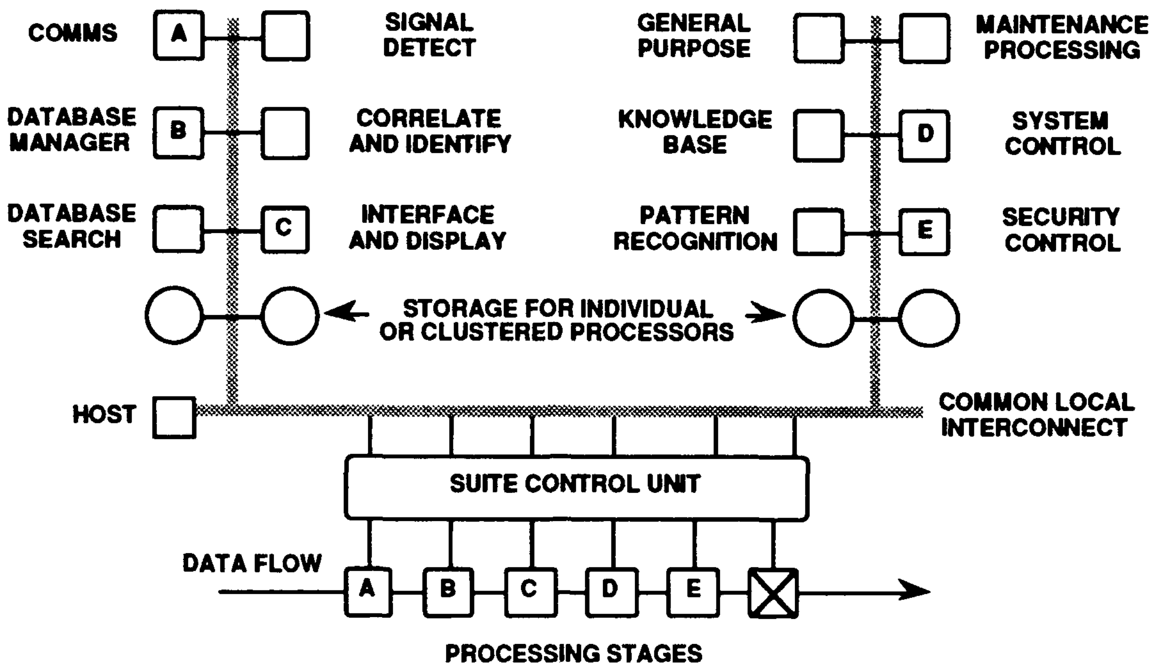



FIGURE 31. VIRTUAL PROCESSING
(MODULAR COMPUTING)

Implicit in this concept is the presence of multiple layers of access to the underlying physical resources. Data management services, for example, may be provided at many levels. A process responsible for maintaining a correlated air track file might access database reports directly, bypassing the higher level query processing and data integrity services of higher level data management functions to achieve minimum time delays. The multiple levels of processing services provided in this concept are analogous to the multiple levels of communications services provided by the Open Systems Interconnection (OSI) model of the International Standards Organization. Just as OSI protocols are being developed to improve interoperability of data communications networks, so can virtual processing resource standards lead to improved interoperability in future computing systems.

In this context, a physical element of a command and control facility will be assembled by configuring first a set of workstations with appropriate embedded special function engines. The workstations are then linked by a high bandwidth local interconnect. More specialized resources then could be attached directly to the interconnect as shared assets.

The notion of a controlled system wide virtual processing environment is illustrated in Figure 32. To adopt such an approach mandates establishment of rigorous controls over the Instruction Set Architectures (ISAs) of the underlying physical resource providers. Given the long service life of tactical command and control systems, it may be necessary to practice configuration control over ISAs in order to maintain integrity of the architecture over an extended period of time and to maintain a viable competitive procurement base for replication or enhancement of system modules.

The NFR-90 design featured an operating system and computer equipment supporting a *virtual machine* or distributed computing architecture. The applications software was to be composed of many separate programs, each operating independently and communicating with other programs through an operating system interface. Communicating applications could not tell, and would not need to know whether the software is all in one computer or in many. The International Schiffs-Studien (ISS) Corporation and participating nations agreed to use this approach.

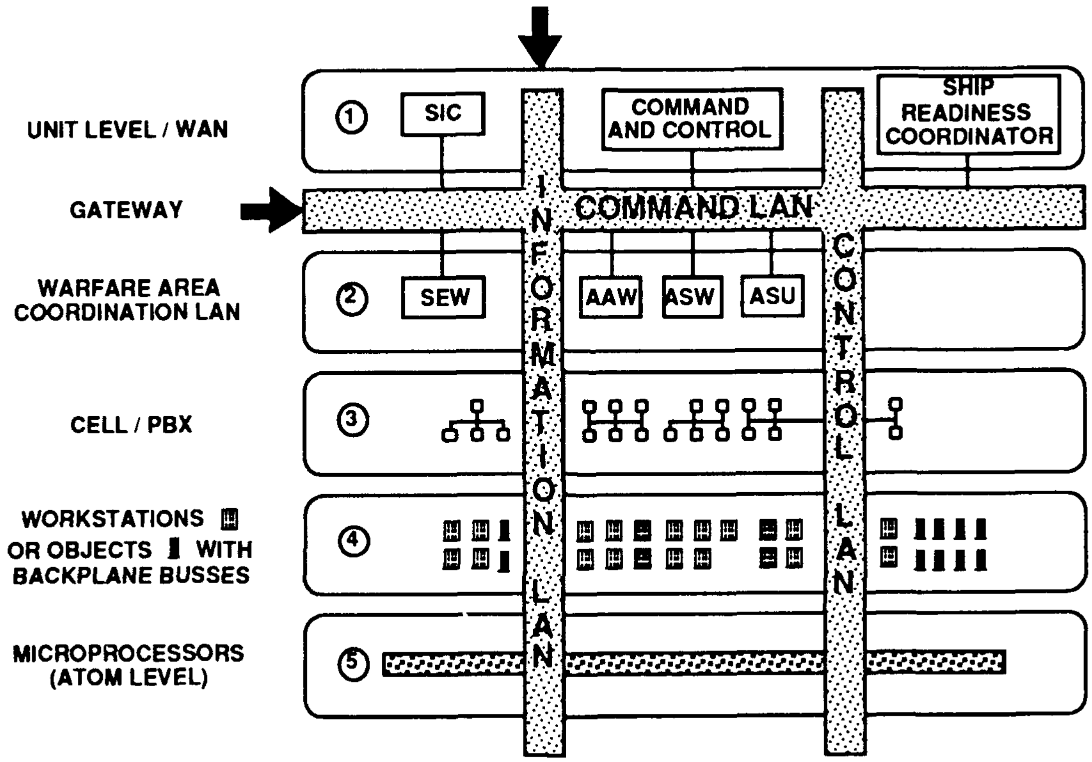



FIGURE 32. LAYERED NETWORK STRUCTURE

## 8.3 DATABASE INTEGRATION

Continued efforts to interconnect and integrate the computers embedded in our combat systems are needed. To a degree, however, the information infrastructure in existing combat systems is also fragmented and disorganized. Major weapon systems may be well organized on an individual basis, but the different approaches used may not fit well together, and then the upper echelons of the battle organization are not well served. What is needed is to form the databases of the different systems into a single, comprehensive database architecture. Though integrated, and consistent with the overall combat system architecture, the solution need not be monolithic: distributed database techniques can be used to break it down into more manageable pieces.

Industry experts suggest that the traditional, short-range approaches to justifying automation investments usually stifle creative projections of what future plants could be like, and that planning should be based on a horizon plant concept that looks ahead at least ten years. Automation will be a key factor in meeting this challenge of global industrial competition, and it is envisioned that the process operations part of a plant will soon run totally under automatic control from startup to shutdown. The Navy is not likely to automate as quickly or completely as industry, but future combat systems will be shaped by the same technical trends. The following is an example of possible horizon plant concepts for combat systems.

Surface combatant lifecycles cover design, implementation, operations, and maintenance phases. As a ship evolves, there is an enormous need to transfer descriptive information between phases. Information is also transferred between people or groups of people within each phase. Today this is done with paper. Traditionally, the documentation is divided into different technical areas, first into ship systems and combat systems, then into the component subsystems. Each set of documentation describes the same ship from a different point of view. The result is that much information is repeated in several different documents; and the updates necessary to make a design change are thus a headache.

Another problem is that, in most cases, today's documentation is implementation oriented and not function oriented. It is tuned for the shipbuilding and construction phase rather than operations.

Computers are already used at every stage in the ship's lifecycle. Today, interconnection of computer systems is a reality and is getting easier all the time. Integration of the computer systems for design, operation, and maintenance is inevitable. But we can go even further and merge the databases of the different systems into a comprehensive ship description database. Though integrated, this common database need not be monolithic; distributed database techniques can be used to break it down into more manageable pieces. With the combined information accessible by computer, different kinds of documentation will be adaptable to different situations. For example, one can imagine maintenance personnel getting a map of the plant, alarm status, operation time, electrical diagrams, and mechanical documentation before they start repair work.

A great deal of knowledge is gained during the design and operation of a surface combatant. Although much information today is transferred between designers and operators, both on paper and via the control systems, this information usually explains how equipment is configured. Seldom is any explanation available to operators about why the equipment is configured in a particular way. The use of automated design tools will make it possible to accumulate knowledge about ship design characteristics and embed such knowledge within its systems. All of this knowledge would be readily available to operators, engineers, and maintenance personnel. Capabilities will be provided for browsing, inquiring, retrieving, and manipulating knowledge. Combat systems designed to the horizon plant concept given here as an example could have the following novel capabilities.

- An open architecture, permitting each ship to tailor configuration of its CIC (or combat system) to specific operating needs. The architecture would include provisions for test and validation of the configuration selected.

- An open architecture, permitting experimentation and testing of new capabilities (e.g., commercial products) during repair and overhaul cycles.

- Ability to *capture* information about plant (combat system) operating trajectories through the use of LANs and related products.

- Full capability for onboard performance and readiness assessment.
- Capability to function as a research, development, test, and evaluation (RDT&E) asset without compromising online systems and mission capabilities.

Data correlation and fusion aids are among the potential future products in this general area. Fusion centers are needed that have capabilities to direct search by onboard and offboard assets; to extract needed information from all available internal and external voice, imagery, and message traffic; and to employ the information for tactical purposes.

## 8.4 COMPUTER PROGRAMS (APPLICATIONS)

Rapid developments in computing, fiber optics, and associated fields such as data fusion promise major advances in combat capability and reliability, primarily by expanded use of microprocessors. Development and updating of computer programs will thus become a dominant activity in all phases of acquisition, demanding major changes in design and procurement philosophy, training, and support.

Since industry is investing heavily in this technology, much more than the defense community at present, development and support costs can be reduced by using commercial standards and products. In particular, commercial sources can be used extensively for such applications as word processing, spreadsheet, text search, database, and graphics.

Customized products are necessary for specialized analysis, secure data processing, and high-resolution imaging functions. For situations where the choice is not obvious, it is important to develop a protocol for deciding when to buy (use commercial sources) and when to build (use government sources) in acquisition of computer resources for combat systems. Since the Navy has a considerable investment in standard computers, displays, test sites, and application programs, the best way to approach this task is not immediately clear. Future combat system developments may emphasize use of community standards for instruction set architectures, backplane busses, LANs, microprocessors, computer programming languages, and development environments.

## 8.5 EXTERNAL COMMUNICATIONS

### 8.5.1 Integrated Communications

The Communications Support System (CSS) will provide for network management, security, standards, and protocols. The CSS integrates and partially automates a ship's external communications. Processing is organized so that dedicated computer programs interface each user and each radio with the system; transfer of data between users and radios is supported by other computer program segments. Future evolution in this area should permit (1) automatic monitoring of own-ship communications status in real-time, and (2) automated allocation of assets, load balancing, and load suppression for own ship to support reconfiguration of battle group/task group tactical networks. Automatic rerouting of communications traffic over alternative circuits (to counter jamming) is included in the latter area.

Further developments along this line appear feasible and useful. This is an area in which automation, driven by industrial needs, can yield better performance while reducing manpower requirements. Eventually dynamic, multifrequency communications capable of handling voice, imagery, and data traffic as well as text must be evolved.

The automated management techniques expected in future communications systems may have secondary utility in anti-C3I operations. Anti-C3I measures involve action to destroy, deceive, disrupt, and/or exploit enemy command systems. The idea is to seize the initiative and achieve first delivery of firepower in effective batches. Communications management aids could be used to assess vulnerability of enemy C3I networks, thus supporting onboard planning and targeting.

### 8.5.2 Network Services

In future tactical operations, when ships, aircraft, and submarines are using military and commercial transmission systems, they are likely to be using communications adapted from OSI design principles. It is necessary to modify OSI designs for the Navy tactical environment, which has many unusual characteristics. These may include narrow bandwidth, rapidly changing error rate performance, units joining and leaving subnetworks without warning, and atypical delays.

### 8.5.3 Communications Servers

Systems in this category provide for automatic routing, distribution control, queueing, and protocol management for message traffic. (In essence, these are gateways to the tactical networks). They also support computer aided message composition, storage, and retrieval. The Naval Communications Processing Afloat Routing System (NAVCOMPARS)/Local Digital Message Exchange (LDMX), Naval Modular Automated Communications System (NAVMACS)/ Common User Digital Information Exchange System (CUDIXS), and Naval Tactical Command System-Afloat (NTCS-A) are examples.

### 8.5.4 Virtual Control Capabilities

Cooperative engagement is defined as a warfighting capability designed to defeat threats through the synergistic integration of distributed resources among two or more units. In fully developed form, cooperative engagement would develop a tactical picture from a wide variety of sensors with sufficient fire-control precision to put the weapon on the target. The basic concept is illustrated in Figure 33. Implicit in such a system is sufficient capability to task sensors, manage the distributed functionality of the network and its processors, and task weapons.

By implication, each unit within the battleforce would be connected to other units within the battleforce by means of a covert, jam-proof, high-capacity network. In broad terms, a fully developed force-wide battle infrastructure is envisioned, capable of enabling the battleforce to be fought under a variety of circumstances with a range of control being effected centrally, on the one hand, to autonomous, on the other. Driving factors that influence the need for such capabilities are listed below:

- First, the emergence of stealth technology is a significant driver for the development and implementation of a cooperative engagement capability for future battleforces.
- A second driver is the emergence of the high-speed, low-altitude seaskimmer and the associated need to extend the horizon for the battleforce.
- Third, greater threat weapon ranges are extending the weapon release line outward, making longer range tactical picture and fire-control quality information necessary.
- Fourth, the emergence of submarines as a launch platform for seaskimming surface to surface missiles or antiair missiles will introduce important new factors in battleforce operations.
- Lastly, the potential for endo-, trans-, and exoatmospheric weapons, such as high flying missiles, transatmospheric platforms, and weapons like tactical ballistic missiles, fractional orbital bombardment, and space-based weapons create new problems for the development of both a tactical picture and needed fire-control data.

Similar forms of *Virtual Connectivity* apply to the STW and ASW mission areas, among others.

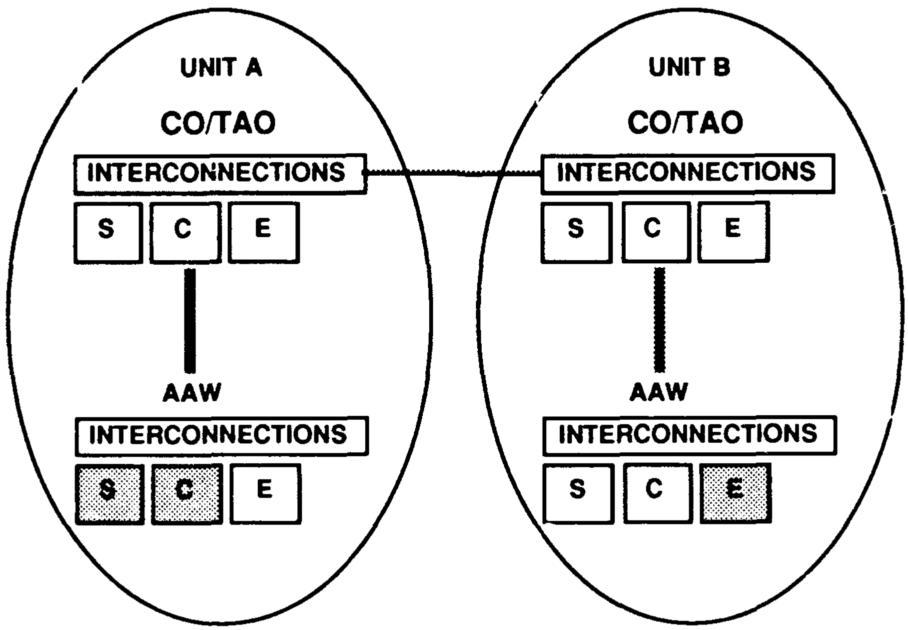



FIGURE 33. AAW COOPERATIVE ENGAGEMENT

## 8.6 SUMMARY

The above areas hold promise for implementation of combat system control strucures in future surface combatants. Both near- and long-term trends have been projected. The principal themes addressed include approaches to backbone control design, extensibility, distributed computing, and continued automation of weapon systems.

# 9.0 SUMMARY: COMBAT SYSTEM DESIGN TRENDS

This section briefly restates important combat system design trends identified in the main body of the report. Many of the key topics are listed in Tables 3 and 4 below. Realization issues and methodology are also considered.

TABLE 3. NEAR-TERM DESIGN TRENDS

| EVOLVING THEMES | BOTTOM-UP MODE |
|---|---|
| Backbone Control Design | Composite Combat System Plant |
| Design for Extensibility | Open Architecture |
| Distributed Computing | Offloading via LAN |
| Continued Automation | Integrated, Reflexive AAW Self-Defense |

TABLE 4. LONG-TERM DESIGN TRENDS

| EVOLVING THEMES | TOP-DOWN MODE |
|---|---|
| Backbone Control Design | Decentralized Control |
| Design for Extensibility | Modular, Layered Combat System |
| Distributed Computing | Virtual Processing |
| Continued Automation | Cooperative Engagement (Virtual Interconnection) |

## 9.1 BACKBONE CONTROL DESIGN

- Synthesis of control structures is a problem of central importance in combat system design, and implementation architectures can be derived from existing theoretical knowledge concerning the control of large scale and complex systems.

- The current cycle of austerity in defense will bring pressure to reduce size, manning, and overall cost of surface combatants. Given the prominence of numbers in base force planning, this could lead to a decline in the proportion of ship acquisition cost devoted to combat systems. Continual improvement in cost/capability ratio (as well as design for extensibility and interoperability) may be a key factor in the struggle to balance warfighting and force planning considerations (capability vs. numbers).

- At present, a bottom-up approach continues to dominate in the design of surface combatants. As existing technology trends mature, however, a transition to top-down combat system design methods is likely. The notion that optimum information structure should drive combat system design, plus the growing technological feasibility of decentralized control structures, will make the problem of backbone control design increasingly important. Distributed computing, LANs, and automatic control technologies are reaching a level of development that could support top-down design efforts within the next decade. The problem is very complex, and the leverage of computer-based design aids should be sought.

## 9.2 EXTENSIBILITY

The rate at which combat systems become obsolete is likely to increase due to changing requirements brought on by the rapidly changing technology of warfare. It must be possible to extend system functionality during the service lifecycle. Further, extension should be possible with a minimum of transitional downtime and without incurring the costs of providing more capability than is needed at the time. Due to the large scale and complexity of the combat system, it should also be possible to employ new or improved components, obtained from a wide variety of suppliers, without causing a series of changes to other components to ripple through the system. This dictates new emphasis on design modularity. Since individual combat systems are elements of higher-level force and theater systems, this entails design for interoperability as well.

- A wide range of future ASW options, together with uncertainty about the evolution of third world submarine threats, argues for increased emphasis on flexibility in surface ship ASW systems. The aim is to permit use of different sensors and weapons within the operating life of a ship. Conceptual design of a flexible backbone ASW system is therefore important. A similar argument can be made for flexibility in AAW.

- The single factor that will have the greatest effect on the conduct of naval operations is the improved flow of intelligence, control, and data signals between all units, including unmanned sensors and weapons. However, future combat system designs will be driven by the action paths made possible by advanced command networks, rather than the command systems themselves. This is especially important for strike warfare, where continued efforts to reduce target detection-to-destruction cycle time are expected.

## 9.3 DISTRIBUTED COMPUTING

The increasing need for combat system integration to meet more stringent control and reaction time requirements will lead to new concepts of data computation, coordination, display, and control, with distributed intelligence provided by a larger number of smaller and multipurpose computers. Development and updating of computer programs may come to dominate the acquisition process, entailing major changes in design and procurement philosophy, training, and support.

## 9.4 CONTINUED AUTOMATION

- Future antiship missiles may achieve speeds of up to Mach 3.5 with sustained high-G maneuvers in both steep diving and very low seaskimming flight profiles. Such missiles have the potential to bypass most layers of current echeloned air defenses, particularly if launched by a submarine at short range. New levels of air defense automation will thus be required. Key factors in countering such threats will include reduced reaction time; increased firepower; and better coordination of sensors, both within and between ships. Fully automated weapon response from detection to destruction will be needed, which involves full integration of all sensors, countermeasures, ECCM, weapons, and C3 systems.

- Determination of combat system warfighting path configurations, together with the coordination of warfighting paths, are driving factors in backbone control design. To a degree, the prevalence of bottom-up design methods reflects the allocation of basic weapon control functions to human rather than automated decision making. The process of automation initiated by the Mk65 gun fire control project in 1945, which evolved into Terrier, Tartar, Naval Tactical Data System (NTDS), and AEGIS remains to be completed. A fully integrated CIC, for example, has not yet been achieved.

- Remotely piloted air vehicles will not replace manned aircraft but will be developed to extend the air capabilities of destroyer types. Potential applications include monitoring of deployed acoustic sensors, radar picketing, radio relay, decoying, and targeting. This will mean increased emphasis on air operations in future combat systems.

## 9.5 REALIZATION ISSUES

- In a broad sense, the overall trend is one of continued automation of an existing hierarchical control structure (the battle organization). The need for consistent evolution underlies the entire approach.

- Overlapping hierarchical control techniques may combine with emerging automation technologies to create new options in reflexive control of self-defense functions.

- Research in the area of model reduction can help us to deal with the different command levels present in combat systems. Deriving architectures from simplified models tailored to unit, warfare area, and warfighting path levels may help to show viewpoint independence of the Vision Architecture. The question of where the designer stands (at the weapon level, looking up; or at the command level, looking down) is significant.

## 9.6 METHODOLOGY ISSUES

- Intuitive methods still dominate in synthesis of process control systems, with the system being specified after the operating process has been designed. An iterative procedure must be used for design because unavoidable uncertainties often make process changes necessary to establish an effective control structure. The problem is also complicated by the scarcity of pilot scale installations where studies can be carried out to evaluate advanced control techniques. Nevertheless, such problems are routinely solved by experienced engineers who have the ability to simultaneously consider:

- Economic, safety and reliability goals of a given process
- Steady state and dynamic behavior of subsystems and units
- Interactions that might occur between control structures
- Failure modes of subsystems and units, including operators
- Possible changes in the process to improve control

These engineers have evolved logical procedures for proceeding from loosely defined flowsheets and goals to well-defined process control systems. Most of these procedures do not involve the use of detailed dynamic models of the process. Reference 49 reports that considerable progress has been made in programming these procedures for computer aided synthesis of process control systems. This research is intended to reduce the cost and development time normally required in engineering complex process plants. Inclusion of synthesis (together with simulation and evaluation) adds greatly to the usefulness of computer aids, particularly when coupled with interactive techniques.

- For large-scale and complex systems, control synthesis methods must be tailored to the area of applications. For example, methods used for design of large chemical process plants differ from those used in design of large systems for communications, transportation, and electrical power distribution. Existing theoretical knowledge is not mature enough to support generic design methods. To improve the scientific and technical knowledge base for combat systems, it follows that empirically based state space modelling and analysis efforts are needed to establish a knowledge base adequate for use in formal combat system analysis and design efforts.

- The overall aim for methodology development is to structure combat system engineering activities so that highly effective and affordable systems are achieved and to identify engineering principles and processes to aid performance of such activities.

- The long-term goal should be to flesh out the idea of a formal combat system architecture model that will link important capability goals and design parameters for a combat system concept with an array of relevant technical and operational knowledge. Knowledge is taken here in a very broad sense including combat system engineering techniques, warfighting doctrine, past designs and their service histories, archives of proven component designs, related cost data and test results, emerging technologies, manufacturing capabilities, and Navy/DOD standard practices. Reference 50 observes that the rapid development of interactive computing has important implications for work on all aspects of automatic control. A successful interactive computing environment can make a specialized body of knowledge more easily accessible and usable, and therefore more easily shared. Given an appropriate interface it also allows an investigator to easily create and modify systems and hence to easily experiment with them. The initial aim probably should be to achieve some level of end-to-end integration of the design process, using approximate methods, rather than to delay integration until a manpower intensive and time-consuming bottom-up engineering process is completed.

- The overall aim for methodology development is to structure combat system engineering activities so that highly effective and affordable systems are achieved and to identify engineering principles and processes to aid performance of such activities. Current approaches to control system analysis and design are fragmented, and there is no accepted design strategy for backbone control systems. Under such conditions, it is important for combat system engineering activities to constantly review their design practices, including deployment of computer aids. Problem drivers include information analysis of design processes, cost structures, analysis of fabrication and assembly practices, and methods for product modularization. With respect to information flows, the idea is to carefully identify the information each design step needs from prior steps or provides to later steps and when that information is needed or available. The steps can then be resequenced to tighten flows of crucial information, producing a faster and more efficient design process. Such reformulations of the design process are very challenging technically and involve defining new work styles, data requirements, and computer support requirements.

## 10.0 REFERENCES


1. Pollard, J.R., Combat Systems Vision 2030: A Combat System Architecture for Future Surface Combatants, TR 91-607, Naval Surface Warfare Center, Dahlgren, VA September 1991.

2. Blanchard, B.S. and Fabrycky, W.J., Systems Engineering and Analysis, Prentice-Hall, second edition 1990.

3. Morari, M.; Arkun, Y.; and Stephanopoulos, G., "Studies in the Synthesis of Control Structures for Chemical Processes: Part I, Formulation of the Problem. (Process Decomposition and the Classification of the Control Stasks; Analysis of the Optimizing Control Structures)," AIChE Journal, 26(2): 220-232, March 1980.

4. Stephanopoulos, G., "Synthesis of Control Systems for Chemical Plants–A Challenge for Creativity," Computers and Chemical Engineering, 7(4): 331-365, 1983.

5. May, R. E., "The Evolutionof Computer Process Control," Control Engineering, pp. 166-173, March 1990.

6. Buckley, P. S., Techniques of Process Control, Wiley, New York, 1964.

7. Douglas, J. M., "Process Operability and Control of Preliminary Designs," 73rd Annual AIChE Meeting, 1980.

8. Douglas, J. M., "A Preliminary Design Procedure for Complete Chemical Plants," Proc. Engineering Foundations of Chemical Process Control (CPC-2), Sea Island, GA, 1981.

9. Shunta, J. P., "Control Systems for Complete Plants: The Industrial View," Proc. Engineering Foundations of Chemical Process Control (CPC-2), Sea Island, GA, 1981.

10. Duren, B. G. and Pollard, J. R., Combat Systems Vision 2030 Combat System Architecture: Design Principles and Methodology, NAVSWC TR 91-795, Naval Surface Warfare Center, Dahlgren, VA 1991.

11. AT&T Quality Steering Committee, Process Quality Management and Improvement Guidelines, Publication Center, AT&T Bell Laboratories, Indianapolis, IN, 1988.

12. Steward, D. V., "The Design Structure System: A Method for Managing the Design of Complex Systems," IEEE Trans. Engineering Management, EM-28(3), August 1981.

13. Constantine, L. L., "The Structured Design Approach," BYTE, pp. 232-233, April 1989.

14. Rouse, W. B. and Cody, W. J., "On the Design of Man-Machine Systems: Principles, Practices and Prospects," Automatica, 24(2): 227-238, 1988.

15. Price, H. E., "The Allocationof Functions in Systems," Human Factors, 27(1): 33-45, 1985.

16. Jamshidi, Modammad, Large Scale Systems: Modeling and Control, North-Holland, 1983.

17. Huff, S. L., "A Methodology for Supporting System Architects During Preliminary Design," Information and Management, 5(1982): 259-268.

18. Tesch, D. and Klein, G., "Optimal Module Clustering in Program and Organization," Proc. 24th Hawaii Int. Conf. on System Sciences, Vol II IEEE Computer Soc. Press, pp. 238-245, 1991.

19. Alexander, Christopher, Notes on the Synthesis of Form, Howard University Press, 1964.

20. Kron, G., Diakoptics, MacDonald, London, 1963.

21. Steward, D. V., "On an Approach to Techniques for the Analysis of the Structure of Large Systems of Equations," SIAM Rev., 4, pp. 321-342, 1962.

22. Steward, D. V., "Partitioning and Tearing Systems of Equations," J. SIAM Numer. Anal. Series B, 2(2):345-365, 1965.

23. Wassel, M. and Wend, H. D., "On the Implementation of Hierarchically Organized Optimization Algorithms," Large Scale Systems Engineering Applications, pp. 110-128, North-Holland, 1979.

24. Lin, Ching-Tai, Structural Controllability, IEEE Trans. Automatic Control, 19(3), pp. 201-208, June 1974.

25. Ho, Y. C., "Team Decision Theory and Information Structures," Proc. IEEE, 68(6), pp. 644-654, 1980.

26. Singh, M. G.; Titli, A.; and Malinowski, K.; "Decentralized Control Design: An Overview," Large Scale Systems: Theory and Applications, pp. 1-15, Pergamon, Proc. of the IFAC/IFORS Symposium, July 1983.

27. Newport, K. T., "Incorporating Survivability Considerations Directly into the network Design Process," IEEE Transactions on Communications, pp. 215-220, 1990.

28. Lefkowitz, I., "Multilevel Approach Applied to Control System Design," ASME Trans. J. Basic Eng., 392, June 1966.

29. Mesarovic, M., "Multilevel Systems and Concepts in Process Control," IEEE Proc., 58, p. 111, 1970.

30. Davison, E. J. and Wang, S. H., "New Results on the Controllability and Observability of General Composite Systems," IEEE Trans. Automatic Control, 20:123-128, February, 1975.

31. Haimes, Y. Y. and Shima, T., "Hierarchical Overlapping Coordination," Systems and Control Encyclopedia, pp. 3564-3568, Pergamon, 1987.

32. Siljak, D. D., "Overlapping Decentralized Control," Large Scale Systems Engineering Applications, pp. 145-166, North-Holland, 1979.

33. Ozguner, Umit, "Decentralized and Distributed Control Approaches and Algorithms," Proc. 28th Conference on Decision and Control, pp. 1289-1294, December 1989.

34. Aoki, M., "On Aggregation of Similar Dynamic Systems," Large Scale Systems Engineering Applications, North-Holland, 1979.

35. Saridis, G. N., "Analytic Formulation of the Principle of Increasing Precision with Decreasing Intelligence for Intelligent Machines," Automatica, 25(3), pp. 461-467, May 1989.

36. D'Azzo, J. J., and Houpis, C. H., Linear Control Systems, McGraw, 1988.

37. Mesarovic, M. D.; Macko, D.; and Takahara, Y., Theory of Hierarchical Multilevel Systems, Academic Press, New York, 1970.

38. Varaiya, P., book review of Mesarovic et al, Theory of Hierarchical Multi-level Systems, IEEE Trans. Automatic Control, 17:280-281, 1972.

39. Fiorio, G. and Villa, A., "A Comparative Analysis and Evaluation of Decentralized Control Structures," Large Scale Systems: Theory and Applications, Pergamon, Proc. IFAC Symposium, pp. 335-339, 1988.

40. Cameron, E. J.; Petschenik, N. H.; Ruston, Lillian; Shah, Swati; and Srinidhi, Hassan, (Bell Communications Research), "From Description to Simulation to Architecture, An Approach to Service-Driven System Development," Proceedings of the First International Conference on Systems Integration, pp. 264-274, April 1990.

41. Calvet, J. L. and Titli, A., "Hierarchical Optimisation and Control of Large Scale Systems With Dynamical Interconnection System," Proc. IFAC Symposium on Large Scale Systems Theory and Applications, Pergamon, pp. 117-126, 1981.

42. Takahara, Y., A Multilevel Structure for a Class of Dynamical Optimization Problems, M.S. Thesis, Case Western Reserve University, Cleveland, OH, 1965.

43. Michaloski, J. and Wheatley, T., System Factors in Real-Time Hierarchical Control, NISTIR-4386, DoC, 1990.

44. Jacklin, S. A., "Arranging Computer Architectures to Create Higher-Performance Controllers," Control and Dynamic Systems, Academic Press, 1988.

45. Schaffer, R. L. and Kloehn, H. G., Design of the NFR-90 Multi-Mission Frigate, Westinghouse Ship Systems, September 1990.

46. Albus, J. S.; Barbera, A. J.; and Nagel, R. N., Theory and Practice of Hierarchical Control, National Bureau of Standards, November 1980.

47. Cullen, R. P., A Concept for Future Force Combat Systems: An Enduring Structure, NAVSWC TR 90-121, Naval Surface Warfare Center, Dahlgren, VA, 1990.

48. Mariton, M.; Bertrand, P.; and Drouin, M., "Large-Scale Systems and Local Area Networks," Proceedings IFAC Large Scale Systems: Theory and Applications, pp. 499-502, 1988.

49. Govind, R. and Powers, G. J., "Control System Synthesis Strategies," AIChE Journal 28(1): 60-73, January 1982.

50. MacFarlane, A. G. J.; Gruebel, G.; and Ackermann, J., "Future Design Environments for Control Engineering," Automatica, 25(2): 165-176, 1989.

# APPENDIX A

# TIME-CRITICAL CONTROL SYSTEMS

# TIME-CRITICAL CONTROL SYSTEMS

This appendix recounts engineering principles for application to time-critical control systems, based on results given by Michaloski and Wheatley.*

## NECESSARY CHARACTERISTICS

In general, time-critical systems are characterized by performance that includes high data communication bandwidth, maximized throughput, and fast execution. This involves the use of fast switching between tasks, small interrupt latency, feature priority scheduling, nonpreemptable status for low priority tasks, and task synchronization. Use of these methods does not, however, define a time-critical system. This motivates the following design principle:

- Reliable and deterministic performance in the time-critical domain are the necessary requirements that define a time-critical control system.

The following corollary design principles are based on the need for deterministic processing time estimates:

- Fairness is not an issue in time-critical control. In applications, either the processes are permanently dedicated to the hardware or are guaranteed to be resident during critical (i.e., nonpreemptable) sections.

- Tasking must be deterministic and user controllable, thus, time slicing is less important. A priority based system where processes execute in known sequence, surrender the processor when finished, or are blocked awaiting an event is preferred.

- Virtual memory is not as important, since the operation of swapping memory in and out from disk is slow and the price of memory is cheap.

- Dynamic creation of processes has limited usefulness, since the overhead required to replace unreliable or crashed processes by new processes is too large.

* J. Michaloski and T. Wheatley, System Factors in Real-Time Hierarchical Control, NISTIR-4836, DoC, 1990.

## CONSTRAINTS IMPOSED BY TIME-CRITICAL DOMAIN

- Simplification improves quality in concurrent as well as sequential computing domains. Simplicity is essential in time-critical controllers to alleviate problems associate with program complexity, timing, and efficiency.

Design methodologies for sequential processing involve the use of small modules, independent modules, black box module definition to hide implementation from purpose, and isolation of logical detail from physical implementation. Much of the structuring is done with information hiding. Structured programming means quality in sequential processing because information hiding simplifies the program modules.

The time-critical aspect of control processing is well served by basic methods of structured programming, but program timing must be assured as well as logical correctness to establish time-critical reliability. The user needs to know not only what the black box will do but also how long it will take and what it will do with unforeseen problems.

- As a design simplifying measure, no module should contain code that waits on an anticipated event. Code that waits on anticipated events can hang any system, sequential or concurrent.

- Use latches, polling, and repeated sampling each control cycle to sample all sensor or real world inputs at a periodic rate, eliminating unpredictable behavior.

## CONCURRENCY AND PARALLELISM

- The major principle governing any effective multiprocessor system design is to exploit the benefits of parallelism while minimizing the impact of parallelism on the algorithms used for processing.

- The basic design principle for interprocessor communication is that it must be efficient enough to justify the additional overhead of communicating between processors, or else the extra processors are extraneous.

- Guaranteed performance (latency rather than throughput) is the ultimate measure of time-critical communication. Latency is defined as the elapsed time before a message is acknowledged. Throughput is the number of bytes one process can send to another in 1 sec.

- In designing system communication protocols, a distinction must be made between cases requiring transfer of small amounts of data every few milliseconds and cases requiring efficient transfer of large amounts of data.

- The concepts of flexibility, data integrity, and extensibility are equally as important in evaluating a communication scheme, but are less tangible.

- Exchange of parameters by message passing (complete copying) offers better flexibility and data integrity in parallel processing than shared memory approaches, but is slower and therefore infeasible for many design constraints (loosely coupled systems).

  Exchange of parameters via shared memory can be very efficient in monoprocessing systems, but in parallel processing it is difficult to assure that all processors have access to the pointers or addresses and that the addresses are valid. Consequently, use of shared memory in concurrent processing systems should be restricted to special purposes such as moving very large amounts of data (or time-critical data) between processes (tightly coupled systems).

COMMUNICATIONS AND CONNECTIVITY

- A server is necessary when one process views communication information logically, while the other process is closely tied to the physical representation.

- From a software validation standpoint, synchronous communication is preferred, but the system may not tolerate the extra amount of overhead.

- One to one synchronous message exchanges, especially for interlevel communication, should be used to encourage a state transition machine that provides a deterministic execution trail.

- Neighboring higher levels should be programmed to delay issuing a new command to the neighboring lower level until the previous message has been acknowledged in the returned status.

- To provide one-to-one correspondence between a command and a status, embed a time stamp within the command and have the lower level acknowledge by returning the current input command with the time stamp embedded within the return status.

Within a message processing system (possible when implemented via shared memory), coordinating the location of the receiver of a message is an important design issue and is termed connectivity of the exchange. Connectivity can be either temporary (datagrams) for the life of the transmission or permanent (virtual circuits). Dynamic connections offer flexibility, but the overhead reduces performance. Static connections are fast, but require the logistical overhead of some centralized server to map the logical to physical locations. The basic design principle applicable to connectivity is as follows:

- Centralize modules that depend on each other and decentralize modules that are independently coupled. In the case of a fault tolerant module, processes that communicate across processors would require dynamic and decentralized connectivity.

## DISTRIBUTION

| | | COPIES |
|---|---|---|
| ATTN | OP 03 | 1 |
| | OP 03C | 1 |
| | OP 05 | 1 |
| | OP 05W | 1 |
| | OP 06 | 1 |
| | OP 07 | 1 |
| | OP 08 | 1 |
| | OP 091 | 1 |
| | OP 094 | 1 |
| | OP 30 | 1 |
| | OP 31 | 1 |
| | OP 32 | 1 |
| | OP 321 | 1 |
| | OP 322 | 1 |
| | OP 324 | 1 |
| | OP 33 | 1 |
| | OP 35 | 1 |
| | OP 37 | 1 |
| | OP 39 | 1 |
| | OP 551 | 1 |
| | OP 603 | 1 |
| | OP 604 | 1 |
| | OP 605A | 1 |
| | OP 70 | 1 |
| | OP 71 | 1 |
| | OP 72 | 1 |
| | OP 73 | 1 |
| | OP 74 | 1 |
| | OP 75 | 1 |
| | OP 76 | 1 |
| | OP 801L | 1 |

CHIEF OF NAVAL OPERATIONS
DEPARTMENT OF THE NAVY
WASHINGTON DC 20350-5000

| | | COPIES |
|---|---|---|
| ATTN | DIRECTOR SURFACE WARFARE | 1 |
| | DIRECTOR AIR WARFARE | 1 |
| | DIRECTOR SUBMARINE WARFARE | 1 |

ASSISTANT SECRETARY OF NAVY (RDA)
DEPARTMENT OF THE NAVY
WASHINGTON DC 20350-5101

| | | |
|---|---|---|
| ATTN | SPAWAR 00 | 1 |
| | SPAWAR 09 | 1 |
| | SPAWAR 005 | 1 |
| | SPAWAR 10 | 1 |
| | SPAWAR 30 | 1 |
| | SPAWAR 30T | 1 |
| | SPAWAR 31 | 1 |
| | SPAWAR 31F | 1 |
| | SPAWAR 314 | 1 |
| | SPAWAR 315 | 1 |
| | SPAWAR 316 | 1 |
| | SPAWAR 317 | 1 |
| | SPAWAR 32 | 1 |
| | PD 40 | 1 |
| | PD 50 | 1 |
| | PD 60 | 1 |

COMMANDER
SPACE AND NAVAL WARFARE
SYSTEMS COMMAND
WASHINGTON DC 20363-5100

| | | |
|---|---|---|
| ATTN | OFFICE OF THE COMMANDANT (CMC) | 1 |

HEADQUARTERS
US MARINE CORPS
WASHINGTON DC 20380-0001

| | | |
|---|---|---|
| ATTN | ONR 100 | 1 |

OFFICE OF CHIEF OF NAVAL RESEARCH
BALLSTON TOWERS
821 QUINCY STREET
ARLINGTON VA 22217-5000

## DISTRIBUTION (CONTINUED)

| | COPIES |
|---|---|
| COMMANDER<br>CARDEROCK DIVISION<br>NAVAL SURFACE WARFARE CENTER<br>BETHESDA MD 20084-5000 | 1 |
| ATTN SEA 00 | 1 |
| SEA 05 | 1 |
| SEA 06 | 1 |
| SEA 06A | 1 |
| SEA 06K | 1 |
| SEA 06D | 1 |
| SEA 06U | 1 |
| SEA 50 | 1 |
| SEA 61 | 1 |
| SEA 91 | 1 |
| SEA 93 | 1 |
| PMS 300 | 1 |
| PMS 312 | 1 |
| PMS 400 | 1 |
| PMS 411 | 1 |
| SEA 441 | 1 |
| SEA 06C | 1 |
| SEA 06 | 1 |
| COMMANDER<br>NAVAL SEA SYSTEMS COMMAND<br>2531 NATIONAL CITY BLDG 3<br>WASHINGTON DC 20362-5160 | |
| ATTN NSWC 00 | 1 |
| COMMANDER<br>NAVAL SURFACE WARFARE CENTER<br>WASHINGTON DC 20362-5101 | |
| COMMANDING OFFICER<br>COASTAL SYSTEMS STATION<br>DAHLGREN DIVISION<br>NAVAL SURFACE WARFARE CENTER<br>PANAMA CITY FL 32407-5000 | 1 |
| COMMANDING OFFICER<br>NAVAL AIR DEVELOPMENT CENTER<br>WARMINSTER PA 18974 | 1 |
| COMMANDER<br>NAVAL UNDERSEA WARFARE<br>CENTER DIVISION<br>NEWPORT RI 02841-5047 | 1 |
| ATTN R LOBB | 1 |
| P DEPOY | 1 |
| CENTER FOR NAVAL ANALYSIS<br>4401 FORD AVENUE<br>ALEXANDRIA AVE<br>ALEXANDRIA VA 22314 | |
| DEFENSE TECHNICAL INFORMATION<br>CENTER<br>CAMERON STATION<br>ALEXANDRIA VA 22314 | 1 |
| WAYNE E MEYER CORP<br>2231 CRYSTAL DR SUITE 1005<br>ARLINGTON VA 22202 | 1 |

INTERNAL DISTRIBUTION:

| | | COPIES |
|---|---|---|
| C | | 1 |
| D | | 1 |
| D2 | | 1 |
| D25 | | 1 |
| E | | 1 |
| E231 | | 3 |
| E232 | | 2 |
| E261 | FINK | 1 |
| E32 | GIDEP | 1 |
| F | | 1 |
| FOX | | 1 |
| F10 | | 1 |
| F20 | | 1 |

# DISTRIBUTION (CONTINUED)

| | | COPIES |
|---|---|---|
| F30 | | 1 |
| F40 | | 1 |
| G | | 1 |
| G10 | | 1 |
| G20 | | 1 |
| G30 | | 1 |
| G40 | | 1 |
| G60 | | 1 |
| G70 | | 1 |
| K | | 1 |
| K10 | | 1 |
| K50 | | 1 |
| N | | 1 |
| N02 | | 1 |
| N04 | | 20 |
| N05 | | 1 |
| N06 | | 1 |
| N10 | | 1 |
| N20 | | 1 |
| N30 | | 1 |
| N35 | CULLEN | 1 |
| N40 | | 1 |
| U | | 1 |
| U04 | | 1 |
| U07 | | 1 |
| U10 | | 1 |
| U20 | | 1 |
| U30 | | 1 |
| U40 | | 1 |

# REPORT DOCUMENTATION PAGE

*Form Approved*
*OBM No. 0704-0188*



**1. AGENCY USE ONLY** *(Leave blank)*

**2. REPORT DATE**
February 1992

**3. REPORT TYPE AND DATES COVERED**

**4. TITLE AND SUBTITLE**
Combat Systems Vision 2030 Conceptual Design of Control Structures for Combat Systems

**5. FUNDING NUMBERS**

**6. AUTHOR(S)**
Bernard G. Duren and James R. Pollard

**7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)**
Naval Surface Warfare Center Dahlgren Division (N04)
Dahlgren, Virginia 22448-5000

**8. PERFORMING ORGANIZATION REPORT NUMBER**
NSWCDD/TR-92/141

**9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)**

**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**

**11. SUPPLEMENTARY NOTES**

**12a. DISTRIBUTION/AVAILABILITY STATEMENT**
Approved for public release; distribution is unlimited.

**12b. DISTRIBUTION CODE**

**13. ABSTRACT** *(Maximum 200 words)*

The centrality of control structures in combat system engineering means that architectures can be derived via existing methods for control of large-scale and complex systems. This report indicates how methods used by system engineers in manufacturing, transport, chemical processing, steel, and distribution systems (for electrical power, water, telecommunications, and resources) can be applied to the problems of combat system engineering. Establishing a theoretical foundation of this type is important because it allows more effective communication and exchange of ideas with experts in both defense and industrial control applications. The underlying purpose is not to invent new theories but to mobilize available scientific and empirical knowledge for application to Navy problems.

**14. SUBJECT TERMS**
Combat Systems Vision 2030, Combat Systems Control Structures

**15. NUMBER OF PAGES**
120

**16. PRICE CODE**

| 17. SECURITY CLASSIFICATION OF REPORT | 18. SECURITY CLASSIFICATION OF THIS PAGE | 19. SECURITY CLASSIFICATION OF ABSTRACT | 20. LIMITATION OF ABSTRACT |
|---|---|---|---|
| UNCLASSIFIED | UNCLASSIFIED | UNCLASSIFIED | SAR |

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. Z39-18
298-102

## GENERAL INSTRUCTIONS FOR COMPLETING SF 298

The Report Documentation Page (RDP) is used in announcing and cataloging reports. It is important that this information be consistent with the rest of the report, particularly the cover and its title page. Instructions for filling in each block of the form follow. It is important to ***stay within the lines*** to meet ***optical scanning requirements***.

**Block 1.** Agency Use Only *(Leave blank)*.

**Block 2.** Report Date. Full publication date including day, month, and year, if available (e.g. 1 Jan 88). Must cite at least the year.

**Block 3.** Type of Report and Dates Covered. State whether report is interim, final, etc. If applicable, enter inclusive report dates (e.g. 10 Jun 87 - 30 Jun 88).

**Block 4.** Title and Subtitle. A title is taken from the part of the report that provides the most meaningful and complete information. When a report is prepared in more than one volume, repeat the primary title, add volume number, and include subtitle for the specific volume. On classified documents enter the title classification in parentheses.

**Block 5.** Funding Numbers. To include contract and grant numbers; may include program element number(s), project number(s), task number(s), and work unit number(s). Use the following labels:

| | | | |
|---|---|---|---|
| **C** | - Contract | **PR** | - Project |
| **G** | - Grant | **TA** | - Task |
| **PE** | - Program Element | **WU** | - Work Unit Accession No. |

**BLOCK 6.** Author(s). Name(s) of person(s) responsible for writing the report, performing the research, or credited with the content of the report. If editor or compiler, this should follow the name(s).

**Block 7.** Performing Organization Name(s) and address(es). Self-explanatory.

**Block 8.** Performing Organization Report Number. Enter the unique alphanumeric report number(s) assigned by the organization performing the report.

**Block 9.** Sponsoring/Monitoring Agency Name(s) and Address(es). Self-explanatory.

**Block 10.** Sponsoring/Monitoring Agency Report Number. *(If Known)*

**Block 11.** Supplementary Notes. Enter information not included elsewhere such as: Prepared in cooperation with...; Trans. of...; To be published in.... When a report is revised, include a statement whether the new report supersedes or supplements the older report.

**Block 12a.** Distribution/Availability Statement. Denotes public availability or limitations. Cite any availability to the public. Enter additional limitations or special markings in all capitals (e.g. NOFORN, REL, ITAR).

| | |
|---|---|
| **DOD** | - See DoDD 5230.24, "Distribution Statements on Technical Documents." |
| **DOE** | - See authorities. |
| **NASA** | - See Handbook NHB 2200.2 |
| **NTIS** | - Leave blank |

**Block 12b.** Distribution Code.

| | |
|---|---|
| **DOD** | - Leave blank. |
| **DOE** | - Enter DOE distribution categories from the Standard Distribution for Unclassified Scientific and Technical Reports. |
| **NASA** | - Leave blank. |
| **NTIS** | - Leave blank. |

**Block 13.** Abstract. Include a brief ***(Maximum 200 words)*** factual summary of the most significant information contained in the report.

**Block 14.** Subject Terms. Keywords or phrases identifying major subjects in the report.

**Block 15.** Number of Pages. Enter the total number of pages.

**Block 16.** Price Code. Enter appropriate price code ***(NTIS only)***

**Block 17.-19.** Security Classifications. Self-explanatory. Enter U.S. Security Classification in accordance with U.S. Security Regulations (i.e., UNCLASSIFIED). If form contains classified information, stamp classification on the top and bottom of this page.

**Block 20.** Limitation of Abstract. This block must be completed to assign a limitation to the abstract. Enter either UL (unlimited or SAR (same as report). An entry in this block is necessary if the abstract is to be limited. If blank, the abstract is assumed to be unlimited.