A NONLINEAR PROGRAMMING METHOD FOR SYSTEM DESIGN WITH RESULTS THAT HAVE BEEN IMPLEMENTED

Frank Hauser
Boeing Aerospace Co.
Seattle, Washington

OPTIMIZATION THEORY

The field of optimization is as broad and interrelated as the problems it attempts to solve. However, there appear to be two general categories:

(1) The indirect or classical methods, based on the calculus of variations, which are often said to involve "parachuting" to the optimum

(2) The direct or programming methods, which involve searching or climbing to the optimum

To list individual methods in the categories is difficult because many that involve indirect methods actually "search for places to parachute". For example, two Lagrangian methods that involve searching are: (1) the Sequential Gradient Restoration Algorithm (ref. 1); and (2) the Projected Lagrangian. The NICO program (Nonlinear Inequality Constrained Optimization) is definitely a Gradient Search Method.

- Indirect or classical, based on calculus of variations (parachuting)
  - Lagrangian methods
  - Optimal control e.g. LQR
  - Etc.
- Direct or programming (searching or climbing)
  - Simplex algorithm
  - Dynamic programming
  - Integer programming
  - Etc.
  - Gradient methods
    - Projected gradient algorithm
    - Method of constrained derivatives
    - NICO
    - Etc.

NICO - GENERAL NONLINEAR PROGRAMMING ALGORITHM

Nonlinear Programming is perhaps the most powerful category in the field of optimization. With it, engineering problems can be solved directly; i.e., the problem does not have to be fit into a canonical form, wherein it can lose some of its features. The general nonlinear programming problem is to determine values for n variables, which satisfy m nonlinear constraints:

$$G_i(X_1, \ldots, X_n)[<, =, >]B_i \qquad i = 1, \ldots, m$$

and, in addition, maximize (or minimize) a nonlinear objective function

$$Z = f(X_1, \ldots, X_n)$$

All engineering problems fit this formulation, though many times the G's and f are only in the engineer's mind as he solves the problem by trial and error. The field of nonlinear programming contains systematic ways to solve the problem if it can be quantified. In most cases, it can be.

NICO can be further classified as a Parameter Optimization method; i.e., it finds the best values for variables whose "first guess" values are input by the user.

Determine X_n values that satisfy nonlinear constraints

$$G_i(X_1, \ldots, X_n) \left\{ \begin{matrix} < \\ = \\ > \end{matrix} \right\} B_i \qquad i = 1, \ldots, m$$

and maximize (minimize) nonlinear objective function

$$Z = F(X_1, \ldots, X_n)$$

Note: n not related to m

NICO ITERATIVELY FORMS AND SOLVES THE FOLLOWING PROBLEM

The fundamental principle in NICO is to iterate to the final solution by solving a succession of linear programming problems.

NICO has two distinctive features: (1) the "first guess" does not have to satisfy any of the constraints (in most real engineering problems, the "first guess" is seldom feasible); and (2) it can converge to local interior as well as exterior optima.

In NICO, the objective is not as important as the constraints. But in most, if not all, engineering problems, the solution is dominated by the constraints. In fact, the optimum is constrained; i.e., constraints literally define the best values of the objective.

- Matrix of constraint equations

$$G_i(\overline{X_o + \Delta X}) = G_{io} + \sum_{j=1}^{n} \left(\frac{\partial G_i}{\partial X_j}\right)_o \Delta X_j \left\{ \begin{matrix} < \\ = \\ > \end{matrix} \right\} B_i$$

Where $\Delta X_j = X_j - X_{jo}$ and $\left(\frac{\partial G_i}{\partial X_j}\right)_o$ is computed via finite difference

- Objective function to be maximized

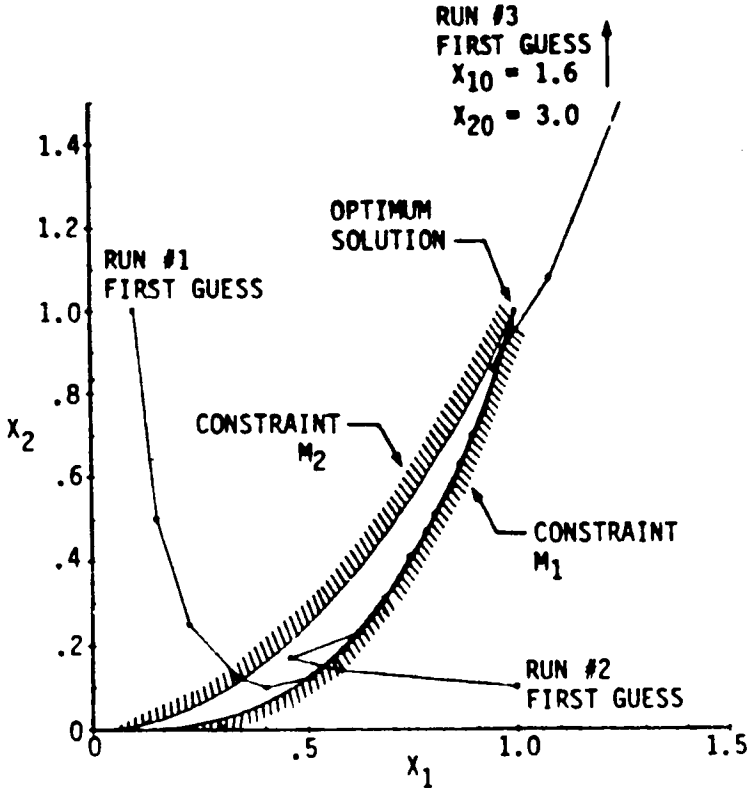
$$Z(\overline{X_o + \Delta X}) = Z_o + \sum_{j=1}^{n} \left(\frac{\partial Z}{\partial X_j}\right)_o \Delta X_j$$

- Variable constraints

$$X_{jL} \leq X_j \leq X_{ju}$$

ACADEMIC OPTIMIZATION EXAMPLE

This figure shows the application of NICO to an academic problem. It shows how NICO iterated to the solution from three different nonfeasible "first guesses". It also shows another distinctive feature of NICO; i.e., it concentrates on the constraints. Run 2 shows how the objective function decreases during the first few iterations in order to reach a feasible solution. NICO moves in a "deflected gradient" direction; i.e., the objective function gradient is deflected by the constraints. Often this is exactly what the engineer does as he optimizes, since most real engineering problems are dominated by the constraints, with the objective function providing only a general direction.



Academic Optimization Example

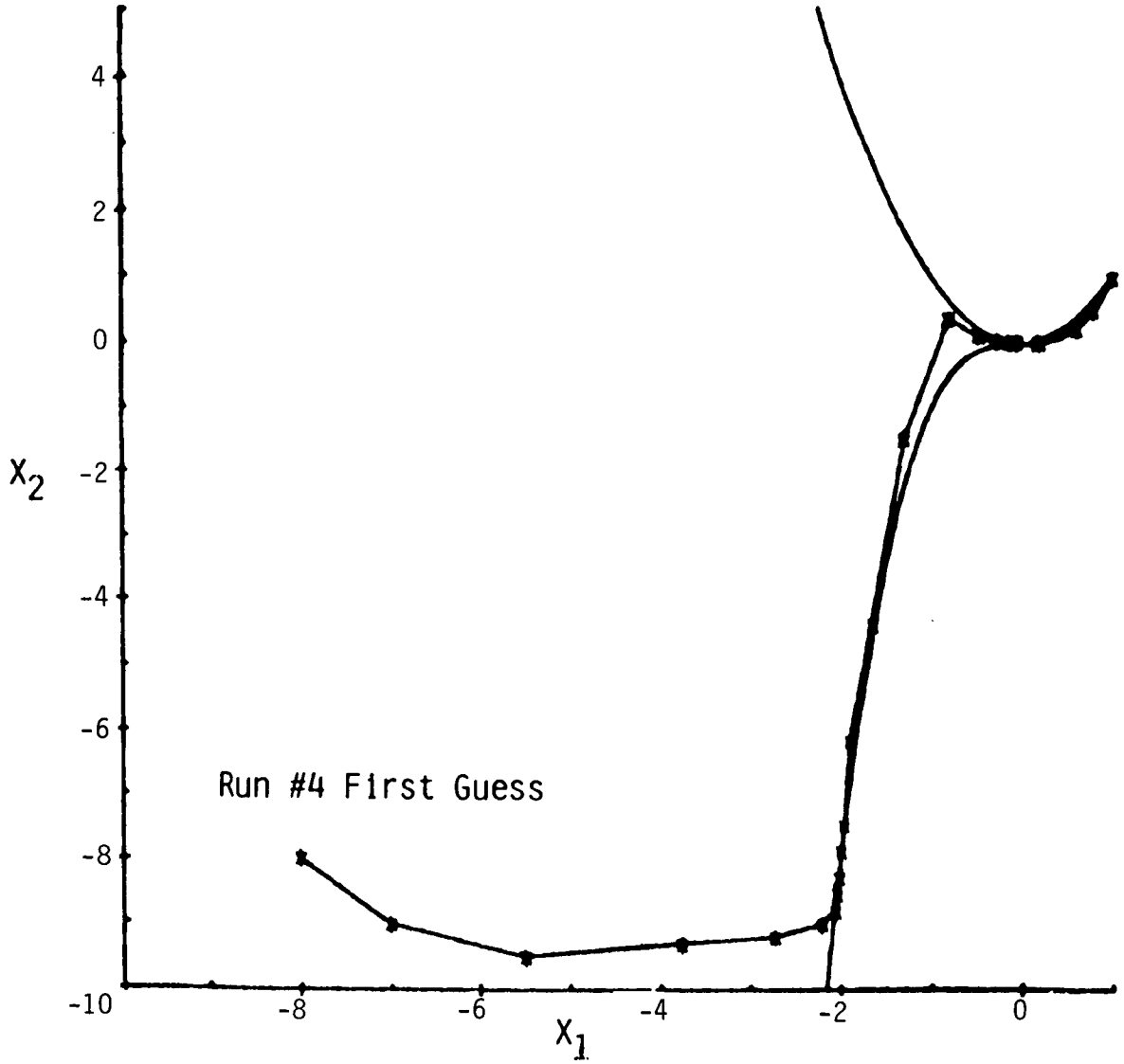
$$\max \quad Z = X_1$$

$$\text{Subject to :} \quad M_1 = X_2 - X_1^3 \geq 0.$$

$$M_2 = X_1^2 - X_2 \geq 0.$$

ACADEMIC OPTIMIZATION EXAMPLE (continued)

The academic example of the previous figure has an unusual situation at the point $X_1 = X_2 = 0$. As shown on this figure, the two "feasible regions" of this example are connected by this single point. NICO had no difficulty passing through this point, because it uses a "hunting" technique and operates much like an engineer as he iterates via a succession of trail points.



Academic Optimization Example

NICO APPLICATIONS

(1) NICO was first applied to launch vehicle autopilot design and produced gains and filters for several (at least two) vehicles that flew. It has also been applied to a so-called Variable Payload Vehicle where it also produced results that were installed in software that actually flew. The latest successful application is on a Variable Trajectory Vehicle, where its design of the roll-yaw autopilot falls in the category of multivariable design.

(2) NICO was also applied to the control effector "trim" or static moment balance problem on an early space shuttle proposal.

(3) Another application included handling quality transient response criteria in the design of a reentry vehicle control system.

(4) Waterjet propulsion and lift system components were "sized" on a large Surface Effect Ship.

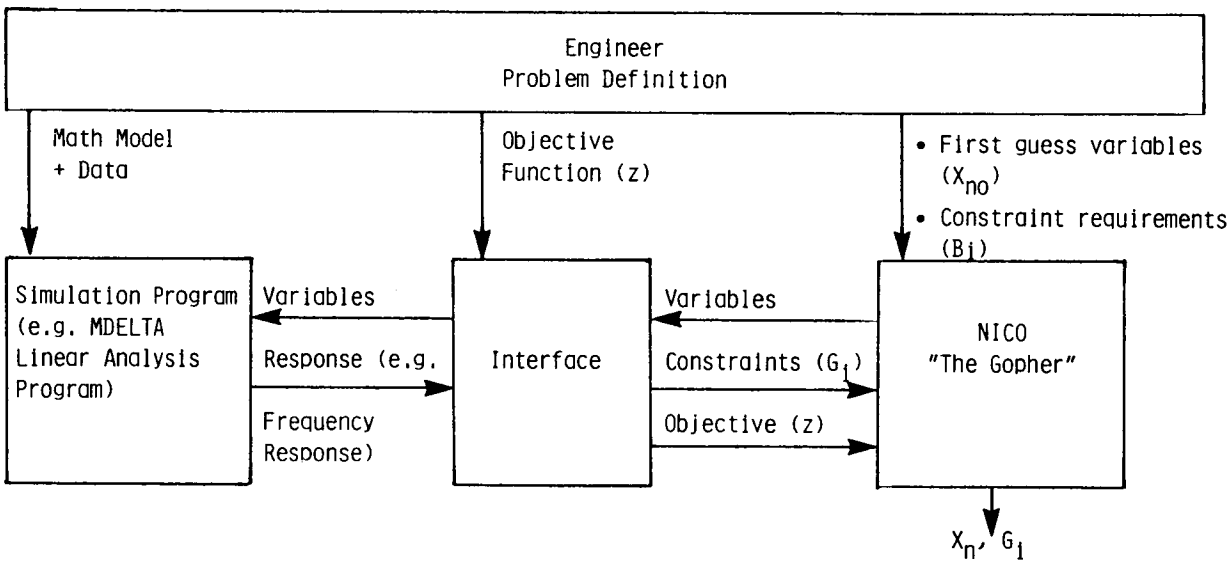
(5) NICO was used in an iceberg transportation study to select candidate icebergs, propulsion system size, and the best route.

• Launch vehicle autopilot design	• Frequency response • Transient response
• Launch vehicle autopilot command mixer design	• Static moment balance
• Reentry vehicle autopilot design	• Handling qualities
• Surface Effect Ship Sizing	• Propulsion and lift system for payload/range
• Iceberg transportation	• Candidate Icebergs, propulsion system, route

NICO LINKED TO LINEAR ANALYSIS PROGRAM

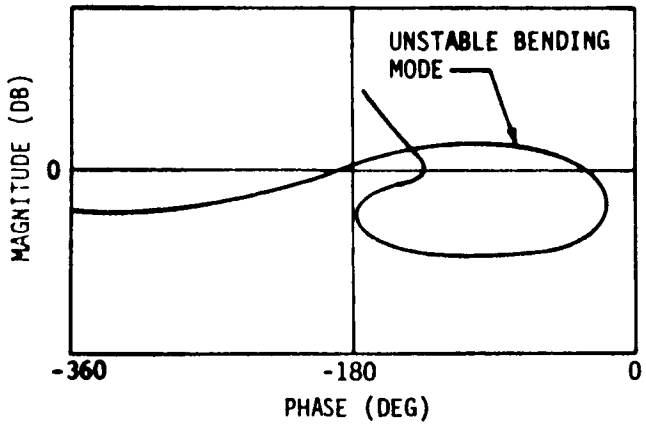
NICO is currently linked as a subroutine to a program that computes frequency responses. Previously, the engineer ran the linear analysis program iteratively, by trial and error selection of control system gains and filters, until the desired stability margins were achieved. NICO replaces the engineer, who now only inputs a "first guess". The engineer can now concentrate on the most important job; i.e., math modelling. NICO does the technically unchallenging job of iterating.

The interface program searches the system response and passes to NICO the array of current values of each of the constraints. It also passes the value of the objective function. NICO passes back new values for the variables that are to be used in computing the next frequency response.

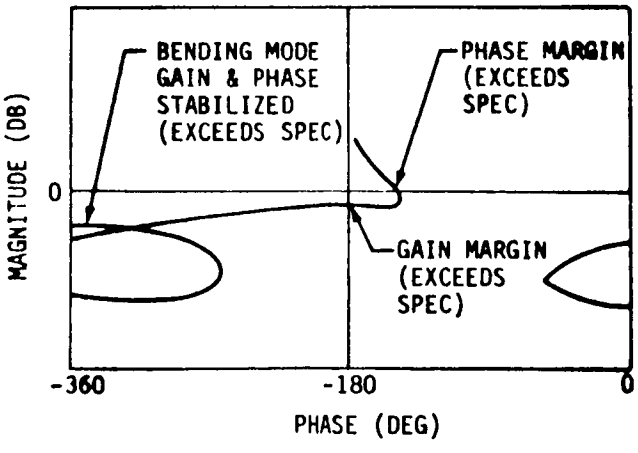


NICO RESULTS THAT HAVE BEEN IMPLEMENTED

NICO was used to design an autopilot for a "so-called" variable payload vehicle. The gains and filters that were computed by NICO were programmed in the software of this vehicle on a mission that actually flew. The top figure shows that the initial user defined (first guess) values for the gains and filters resulted in an unstable bending mode. After six iterations, NICO produced values that stabilized this mode and exceeded all required stability margins. The final iteration resulted in a better system than had been previously designed by manual techniques, and it is felt that NICO required less engineering and computer time.



- FIRST GUESS RESULTED IN UNSTABLE BENDING MODE



- 6 ITERATIONS LATER, STABILIZED WITH ALL MARGINS MET

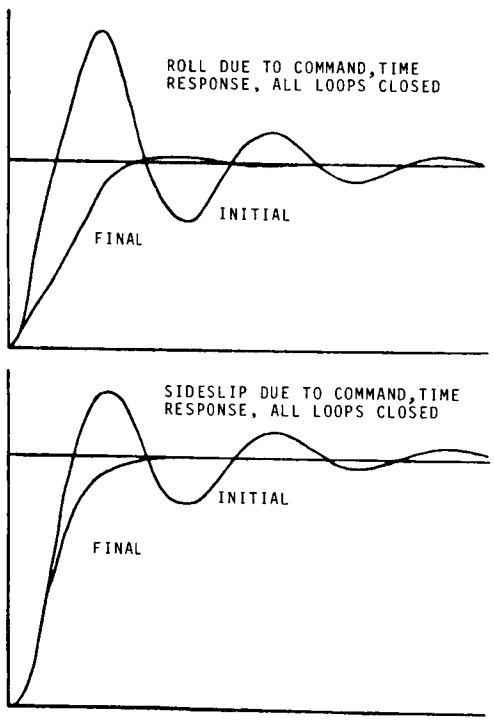
NICO MULTI-INPUT MULTI-OUTPUT CONTROL SYSTEM DESIGN

There have always been discussions about the shortcomings of classical frequency response design methods when dealing with multi-input multi-output (MIMO) control system design. The difficulties with MIMO systems arise from interactive effects; i.e., the action of one feedback loop affects the actions of the others. The standard practice used successfully by the "classical" engineer is simply to design all loops simultaneously. This iterative design approach includes an extensive tolerance analysis and yields optimum systems. NICO automates this classical approach.

This figure is a remarkable example of NICO applied to MIMO. This was the design of a roll-yaw autopilot on a variable trajectory vehicle. On every step, NICO simultaneously considered:

(1) Stability margins of the roll loop with the yaw loop closed
(2) Stability margins of the yaw loop with the roll loop closed
(3) Shape of the frequency response of roll due to command with all loops closed
(4) Shape of the frequency response of sideslip due to command with all loops closed

Disturbance rejection could also have been simultaneously considered by reducing the peak value of the frequency response of roll, sideslip, and all fin angles due to a disturbance like the wind.



PHASE MARGIN (DEGREES)		
	INITIAL	FINAL
ROLL OPEN, YAW CLOSED	22.	60.

FREQUENCY RESPONSE, PEAK RESONANCE (dB), ALL LOOPS CLOSED		
	INITIAL	FINAL
ROLL DUE TO COMMAND	9.	0.
SIDESLIP DUE TO COMMAND	5.	0.

SURFACE EFFECT SHIP SIZING

At one point in its development, NICO was linked to a program that computed the range/speed/payload performance of a Surface Effect Ship. NICO was used to establish: (1) the pump diameter, inlet area, and nozzle area of the waterjet propulsion system; (2) the lift system air flow rate; and (3) the vehicle speed that would maximize the payload for a given range. The constraints were: (1) maximum pump diameter, inlet area, nozzle area, horsepower, and air flow rate; (2) minimum speed; and (3) maximum speed possible while avoiding pump cavitation.

RESULTS

	Engineer	NICO	Change
• Payload (short tons)	3760	4000	+6.4%
• Cruise Speed (knots)	43 (cavitating)	46 (No cavit.)	+6.5%
• Travel Time (hours)	60	56.7	+5.4%

ADVANTAGES

	Engineer	NICO
• Faster Response	Over two weeks of engineering manhours	1 day
• Lower Cost	Over 40 runs varying major parameters (>$100)	1 run (< $20)
• Improved Results	Feasible ship found, but not true optimum	True Optimum

REFERENCE

1. Miele, A.; Calastro, A.; Rossi, F.; and Wu, A. K.: A Modification of the Sequential Gradient Restoration Algorithm for Mathematical Programming Problems with Inequality Constraints, Aero-Astronautics Report No. 124, Rice Univ., 1975.