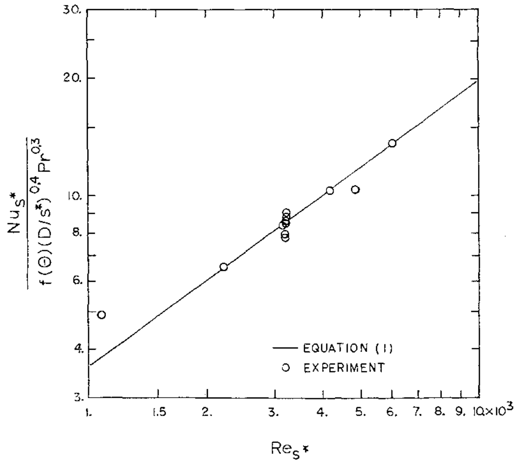

Fig. 2 Correlation of test results.

The local heat-transfer coefficient may be obtained approximately by the following procedure. It was found that the local heat-transfer coefficient distribution follows a Gaussian probability density distribution curve. The results may be expressed as

$$h_l = h_{\min} + (h_{\max} - h_{\min})e^{-\eta^2/2} \tag{6}$$

For air it was found that over all the tests the maximum and minimum heat-transfer coefficients could be approximately expressed as

$$h_{\max} = 1.40\, h_{\text{ave}} \tag{7}$$

$$h_{\min} = 0.70\, h_{\text{ave}} \tag{8}$$

$h_{\max}$ is the heat-transfer coefficient found at the point of impingement. $h_{\min}$ is found at extreme end of the surface furthest removed from point of impingement. h_{ave} is obtained from Eq. (1). The exponent may be expressed as

$$\eta = 2.4\, X/L \tag{9}$$

X is measured from the point of impingement along the surface.

Conclusion

This series of tests has provided a usable correlation for the design of systems using impinging air jets on concave enclosed surfaces. The empirical correlation, Eq. (1) suggests a Nusselt number dependency on the Reynolds number to a 0.7 power. This compares quite well to the available literature, which is almost solely for flat plate surface geometries. References 1 and 3–7 give Reynolds number exponents in the range from 0.6 to 0.8. This seems to imply that if the surface geometry is concave, so long as it is smooth and continuous, the jet air mass flow rate affects the heat-transfer coefficient in the same manner as in the flat plate case. The distance from jet exit to surface spacing affects the Nusselt number to a -0.4 power. This compares to a -0.6 power as found in literature.[7] This result shows that the jet impingement heat transfer in a closed cavity is affected to a lesser degree by the distance between surface and jet exit than for a flat plate surface case. This implies that the gas jet tends to re-entrain more of its own mass, thus causing recirculation currents within the cavity. This effect tends to dampen the effect of jet nozzle distance from the surface. The angular effect is related by $f(\theta)$. This parameter tends to give a lesser degree of effect in concave surfaces than in flat plate geometries. The reason for this behavior is again due to the existence of recirculation currents within enclosed cavities and the fact that the distance from the jet nozzle to the surface does not change to the same degree in an enclosed concave cavity as in the flat plate case.

Table 1 Range of applicability of correlation, Eq. (1)

Variable	Range
Re_s*	1000–8000
$D/S*$	50–120
θ	$-20°$–$20°$
$Sp/d*$	2.5–10
$L/S*$	500–700

References

[1] Smirnov, V. A., Verevochkin, G. E., and Brdlick, P. M., "Heat Transfer Between a Jet and a Held Plate Normal to Flow," *International Journal of Heat and Mass Transfer*, Vol. 2, Pergamon Press, New York, 1961, p. 1.

[2] Jusionis, V. J., "Heat Transfer Effects of Impinging Air Jets on an Enclosed Concave Surface," LR 22312, March 1969, Lockheed-California Co., Burbank, Calif.

[3] Perry, K. P., "Heat Transfer by Convection from a Hot Gas Jet to a Plane Surface," *Proceeding of the Institution of Mechanical Engineers*, Vol. 168, No. 30, 1954, p. 775.

[4] Gardon, R. and Cobonpue, J., "Heat Transfer Between a Flat Plate and Jets of Air Impinging on It," *International Heat Transfer Conference*, 1961, p. 454.

[5] McMurray, D. C. et al., "Influence of Impinging Jet Variables on Local Heat Transfer Coefficient Along a Flat Surface with Constant Heat Flux," *Third International Heat Transfer Conference*, 1966, Chicago.

[6] Schuh, H. and Petterson, R., "Heat Transfer by Arrays of Two-Dimensional Jets Directed Normal to Surfaces Including the Effects of a Superposed Wall-Parallel Flow," *Third International Heat Transfer Conference*, 1966, Chicago.

[7] Akfirat, J. C., "Transfer of Heat from an Isothermal Flat Plate to a Two-Dimensional Wall Jet," *Third International Heat Transfer Conference*, 1966, Chicago.

TACTICS—a Three-Body Simulation Program

Jerome H. Hutcheson*

The RAND Corporation, Santa Monica, Calif.

Introduction

TACTICS is a computer program for use in simulating the kinematics and dynamics of motion of three vehicles in three-dimensional space. Care has been taken to make the program versatile so that it may be used in studing a wide variety of problems; but, in keeping with the program's initial purpose, the most important capabilities relate to interceptor-target guidance and to intercept trajectories in general. Since the flights of three separate vehicles may be represented simultaneously, the program can be used to simulate aerial combat between aircraft (e.g., a two-on-one engagement or a one-on-one with missile launching). On the other hand, there are a number of other possibilities, such as 1) using one or more of the vehicles to represent an ASM, SAM, or SS, 2) having the target represent an ICBM reentry vehicle, 3) using vehicles 1 and 2 to represent first- and second-stage boosters, or 4) having one vehicle represent an orbiting satellite.

Presented as Paper 69–890 at the AIAA Guidance, Control, and Flight Mechanics Conference, Princeton, N. J., August 18-20, 1969; submitted August 11, 1969; revision received September 25, 1969. This research is supported by the U.S. Air Force under Project RAND—Contract No. F44620-67-C-0045.

* Directorate of Operational Requirements and Development Plans, Deputy Chief of Staff, Research and Development, U.S. Air Force.

So far, the model has been used primarily in simulating fighter vs fighter and missile vs fighter duels. Therefore, its development is further advanced in this area at present, because of the number of maneuver routines created for this specialized purpose. However, ASM and SAM simulations have been performed in connection with other projects, and the program has been checked out for satellite applications. Its usefulness and potential increase with each new application.

TACTICS was developed primarily as a research tool to explore in detail the mechanics, geometry, and vehicle performance characteristics involved in an interceptor-target engagement. The output is a step-by-step time history of variables relating to the position, velocity, acceleration, applied forces, vehicle attitude orientation, and aerodynamics of each of the three vehicles.

Miss distance or point of closest approach between two of the three vehicles may be calculated, and the problem run may be automatically terminated if so desired. The program has been built in modular building-block form in order to allow the desired flexibility and growth potential. The construction is, in many respects, comparable to that of the ROCKET program[1]; the important differences are inherent in their different purposes. In formulating TACTICS, many ideas were borrowed from ROCKET, including its most important distinguishing feature: that of being a general trajectory program which allows the researcher to obtain results without recourse to a detailed knowledge of the inner workings of the program.

Scope

The program is able to simulate the flight or trajectory of almost any type of vehicle in almost any combination. A variety of guidance-law subroutines is available for simulating terminal homing or command-guided trajectories (e.g., biased-proportional and proportional-navigation, lead-collision, pursuit and lead-pursuit trajectories). There is also a variety of open-ended control-law subroutines available for simulating aerodynamic maneuvers such as turning, diving, climbing, and combinations thereof. The library of guidance- and control-law subroutines has been growing and is likely to continue to do so as new problems are encountered. In fact, it has proved to be very convenient to package the unique aspects of a particular vehicle or problem into one of these control-law subroutines.

There are options available for considering the earth as flat or round, fixed or rotating, but the gravitational field is a simple inverse square law field.

Associated with each of the three vehicles are six degrees of freedom, three translational and three rotational, the latter defining orientation or attitude. The vehicles may be either fixed or in motion with respect to the earth.

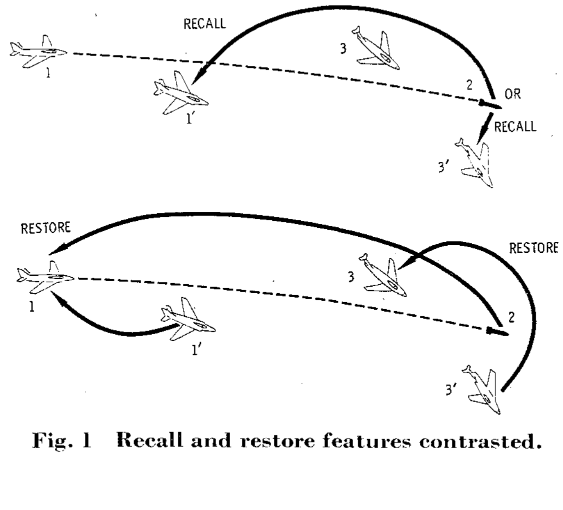

Fig. 1 Recall and restore features contrasted.

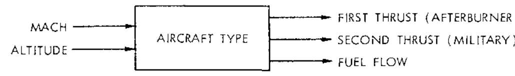

Fig. 2 Basic framework for flight simulation (generalized).

As previously indicated, the step-by-step time history of the engagement is limited to the consideration of the performance of three vehicles at one time. However, two devices have been incorporated for extending this performance capability to more than three vehicles by using sequential computations. These devices are designated "recall" and "restore" and are illustrated in Fig. 1. Consider a vehicle 1 which launches a missile 2 at a target 3. At some subsequent time or event (e.g., a miss), it may be desirable to recall 2 or place it in captive flight on either 1 or 3 for a subsequent quasi-vehicle 4. (It is not necessary that 4 have the same characteristics as or even resemble 2.) To illustrate the "restore" feature, the same example as above has been used except that at some time or event subsequent to the launching of vehicle 2 (e.g., hit, miss, or ground impact) we wish to restore the situation exactly as it was at launch time. After restoration has occurred, events may proceed and a new launch (4, 5, etc.) may take place at a subsequent time or event. On the other hand, branching may be desired whereby some characteristic or parameter is altered, and 4, 5, etc., are to be launched under the same restored conditions of time and geometry.

Basic Framework

In organizing TACTICS, the prime emphasis was placed on providing versatility to accommodate a broad spectrum of problems. At the same time, the program would ideally be easy to use and conveniently adaptable to refinements and new features. Accordingly, it was built in modular building-block form, incorporating many subroutines, which at the choice of the user could be replaced, modified, or simply not employed. In fact, new building blocks are welcomed for increasing the potential problem-solving capability.

The following procedures vital to any program for simulating flight trajectories are to 1) input initial-condition data, 2) calculate the geometry, 3) specify the applied forces involved, 4) integrate the equations of motion, and 5) output the results. These procedures, shown schematically in Fig. 2, constitute the basic framework of TACTICS to which embellishments are added (e.g., model atmosphere, coordinate transformations, aerodynamic computations, etc.). Although several options are provided for modes of integration and for output form, this basic framework may be considered inflexible except for specifying the applied forces. In this last respect, maximum flexibility is provided by an arrangement comparable to the employment of plug-in modules. That is, the applied forces, and the time(s), event(s), or situation(s) dictating their application, are contained within two subroutines which may be specialized to deal with a particular problem. To illustrate, a particular problem is defined by initial-condition input data and by a number of POLICY statements (subroutine). These statements are logical expressions which dictate the control laws (subroutines) governing the flight of each vehicle. They are usually conditionally based on time or on geometrical and kinematic relationships.

TACTICS: a Simplified View of Its Operation

A rudimentary description follows of the operational principles involved in simulating an intercept problem. Imagine a vehicle in three-dimensional space which has a position and a velocity as defined by initial-condition input data. If no accelerations or net applied forces are involved, the time history or trajectory of the vehicle will obviously be a straight line. However, in the general trajectory case, there will be

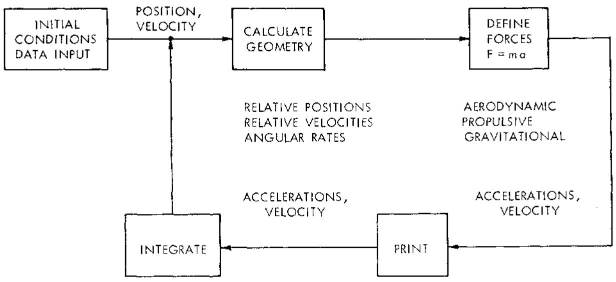

Fig. 3 Input-output form.

a net acceleration $(\ddot{x}, \ddot{y}, \ddot{z})$ due to gravitational, aerodynamic, and thrust forces. Assuming that all forces can be defined and specified, the trajectory problem or simulation is reduced to one of integrating the net acceleration as a function of time to obtain velocity, and of integrating velocity to obtain position. If the applied net force were constant or a simple function of time, the trajectory simulation would be straightforward and relatively simple. In the general case, particularly where closed-loop guidance and aerodynamics are involved, the forces may be complicated functions of position, velocity, geometry, and time [as well as of the behavior or predicted behavior of some other vehicle(s)]; hence, numerical integration techniques must be used to calculate the trajectory stepwise. It should be clear from the preceding discussion that in order to start a problem run, input data for initial position and velocity must be supplied. The bulk of all other input-data requirements will pertain to parameters associated with the definition or calculation of the forces which are to be applied (e.g., lift, drag, thrust, and gravitation). The intercept-problem definition, written in terms of logic, timing, and/or geometrical events which dictate what type of control laws are to be applied and under what circumstances, is contained in a POLICY subroutine (which in a sense is also an input to the program). The calculation of the forces or force functions which are to be called upon by POLICY are contained in the library of control-law subroutines. It is not expected that this library will ever include every conceivable force function or guidance law, especially since TACTICS is a research tool for experimentation. However, experience has shown that new subroutines may usually be conveniently generated by modifying those subroutines already on hand.

Defining the Forces

Control laws

TACTICS integrates the equations of motion defined by the three components of net acceleration associated with each of the vehicles. A *resultant* vector force **F** will define a net vector acceleration **a**. The forces of primary concern which will add up to this resultant force **F** may be categorized as 1) gravitational, 2) aerodynamic lift and drag, and 3) propulsive. If we assume that these forces are defined in magnitude and direction, it is a straightforward procedure to add them up, resolve them into components, and determine the net acceleration. Each control-law subroutine may be considered as a modular unit where this process or its equivalent is performed, the output being the three components of net accleration applicable to a particular vehicle.

There are many control laws where the net acceleation is by definition the starting point, and all control-law computations are concerned with finding the correspondence between the forces which would be necessary to create such an acceleration.

For example, consider a hypothetical control law where all components of net acceleration are to be zero. Assume that propulsive and gravitational forces have been determined. The function of the control law in this simple case is to determine the aerodynamic lift and drag forces necessary to guarantee the postulated output condition, i.e., zero acceleration.

The three components of net acceleration are resolved as follows: 1) turning acceleration in the horizontal plane, 2) climbing or diving acceleration in the vertical plane, and 3) change in speed or magnitude of the velocity.

Because of real-life considerations, it is often necessary, in simulating guidance of aircraft or missiles, to modify commanded values because of constraints such as structural or aerodynamic limitations, time lag, and noise. Accordingly, let us assume that the lateral acceleration components may be operated upon so that modified values will result. The third component, change in speed, is taken to be only a function of propulsion and aerodynamic drag forces with no constraints.

Some of the basic guidance and control laws which are used in TACTICS are listed as following: 1) right or left turn, 2) straight flight, 3) climb or dive, 4) proportional navigation, 5) pursuit-course navigation, and 6) missile (Sidewinder, Falcon, etc.). There are many other variations or combinations which are too specialized to describe here. One important point should be emphasized: each law, as described, is, in the general case, subject to aerodynamic ($CL_{\max}$ or $\alpha_{\max}$) and structural ($a_{s_{\max}}$) constraints or limits. Accordingly, if one of these boundary values is reached, the control law is automatically altered to accommodate for the limit condition.

Many of the significant guidance and aerodynamic parameters vary with Mach number and are unique to specific missile designs. This is also true for the boost-burn-guide time intervals. Accordingly, these detailed characteristics are packaged into specialized subroutines.

Aerodynamic forces

As mentioned previously, aerodynamics forces are usually important factors in defining the components of acceleration and, in particular, changes in speed. The computations may involve a large number of input parameters corresponding to the flight characteristics of a particular airframe. On the other hand, there may be problems where aerodynamic forces are not even taken into consideration, or are considered only in an elementary form.

In simulating the flight of specific types of aircraft design, provisions have been made for incorporating tables to describe the interactions among lift coefficient C_L, drag coefficient C_D, angle of attack α, maximum lift coefficient $C_{L_{\max}}$, thrust fuel flow, and Mach number. Interpolation routines are employed to provide the equivalence of a functional relationship.

An alternative procedure to using tables is to treat the aerodynamic parameters as variables, e.g., functions of Mach number, and to use analytic functions for determining the forces. Provisions have been made in TACTICS for using the aerodynamic functional relationships usually applicable. If vacuum flight is to be assumed, all aerodynamic calculations may be eliminated.

Propulsive forces

The parameters and values associated with propulsive forces are handled in almost the same way as those associated with aerodynamic forces. In simulating the flight of certain aircraft types, provisions have been made for incorporating tables, and interpolation routines are used. The input-output form is represented in Fig. 3. Alternative procedures analogous to those described above for aerodynamic forces are as follows: 1) other tables may be prepared using the same format, 2) analytic functions may be defined within a particular control law, and 3) constant values for military and afterburner thrust may be read in as input initial-condition data.

Gravitational forces

TACTICS is automatically set for the simplest assumption of a gravitational force—a flat earth with 1 g acting downwards in the negative z direction. However, options for considering the more complicated cases of a rotating or nonrotating round earth have been provided so that centrifugal, coriolis, and inverse-square law effects may be included if pertinent to the problem.

Program Inputs

POLICY subroutine

This portion of the program is supplied by the user. It consists of logical statements and expressions which dictate the guidance or control-law subroutines governing the flight of each vehicle. These subroutines define the various forces which are to be applied to a vehicle to perform a certain maneuver or to guide in accordance with some prescribed guidance law. The resultant forces will define a net acceleration. In order to write a POLICY routine, familiarity with the mathematics involved in guidance, aerodynamics, propulsion, etc., is not essential. However, many options are available, and the user should be familiar both with a dictionary for selecting these options and with the library of available control or guidance-law routines. The use of a POLICY subroutine is shown in the following example:

Vehicle 1 (F-104 intercepting aircraft) 1) Fly lead-collision navigation course. Thrust: military. 2) If range to target is less than 6500 ft, and angle off target's tail is less than 30°, launch missile. 3) Pull constant Mach, constant-altitude right turn. Thrust: afterburner.

Vechile 2 (proportional navigation missile) 1) Fly "captive flight" until above launch criteria are satisfied. 2) Launch, boost, fly unguided and then guided in accordance with guidance and aerodynamic characteristics specified in special missile subroutine. 3) When range rate (missile-target) becomes greater than zero, initiate process for finding miss distance and end program.

Vehicle 3 (F-105 target aircraft) 1) Fly striaght and level. Thrust: 80% throttle setting, military power. 2) When missile is launched, perform 5.5-g constant-Mach left turn. Thrust: afterburner.

Criteria and options

Considering the preceding illustration, the most obvious example of choosing an option is in the selection of the control-law routines. In the argument listings for these routines there are also options pertaining to 1) whether the law is to govern vehicle 1, 2, or 3, 2) the magnitude (and perhaps direction) of the propulsive or thrust force, 3) the number of g's commanded for the climb, dive, or turn maneuvers, and 4) whether tabulated values or analytic functions are to be used in calculating aerodynamic forces.

So far this subject has been discussed in the context of choices to be made within the POLICY routine. There are numerous other options which may be selected by reading in flags or constants as part of initial-condition data. However, it is sometimes desirable to over ride these initial instructions in POLICY if some certain situation arises during a problem run which requires, perhaps, a change in frequency of printout or integration-step size.

Summary and Conclusions

The TACTICS simulation program was developed primarily as a research tool for studying interceptor-target trajectories. The development is considered complete in terms of basic framework, organization, input-output integration, flow, etc. However, in line with its purpose as a research tool, it is open-ended, subject to adaptation for each new problem, and, in this sense, never complete. This adaptive process is believed to be simple and flexible because of the options available; it is important, also, because the main variables defining a problem can be handled externally by modular units (control-law and POLICY subroutines). Hundreds of simulation problems have been run under a wide variety of input-data, control law, and POLICY options. These runs have involved the simulation of air-to-air combat, SAM, and ASM missile applications. Experiments have also been performed for space applications using the two-body equation of motion. Finally, there is no guarantee that the program will work perfectly for all cases in spite of all the checkout and operational experience. The number of possible configurations and combinations is extremely large, and it is unlikely that all will ever be tried. As with all computer results, skepticism, intuitive reasoning, and cross checking are in order.

Reference

[1] Boehm, Barry, "ROCKET: RAND's Omnibus Calculator of the Kinematics of Earth Trajectories," RM-3534-PR, July 1963, The RAND Corp., Santa Monica, Calif.

Secondary Jet Interaction with a Subsonic Mainstream

ROBERT ROSEN* AND LOUIS A. CASSEL†
McDonnell Douglas Astronautics Company, Western Division, Santa Monica, Calif.

Nomenclature

A = arbitrary constant
r_j = radius of jet exit
r_o = effective jet radius
U_j = jet velocity at exit
U_∞ = mainstream velocity
C_p = $(p - p_\infty)/q_\infty$, pressure coefficient
q_∞ = freestream dynamic pressure
$\bar{r}$ = radial distance from nozzle centerline normalized by r_o
ω = azimuthal angle

SEVERAL analyses exist in the literature concerning the interaction flowfield resulting when a secondary jet exhausts transverse to a uniform stream. In those works, two classes of problems have been considered: the VTOL transition problem involving a subsonic mainstream and low

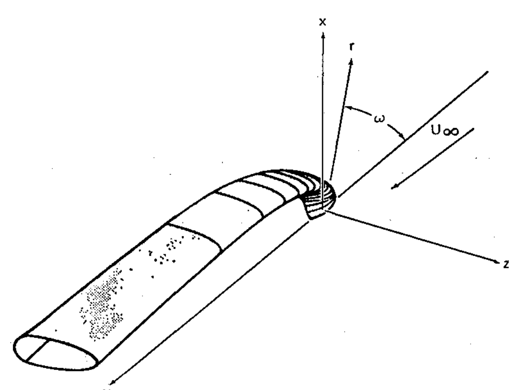

Fig. 1 Coordinate system.

Received September 8, 1969; revision received October 27, 1969. This work was performed under Contract DAAH01-68-C-1919 with the U.S. Army Missile Command.

* Senior Engineer/Scientist. Member AIAA.
† Supervisor, Advanced Controls. Member AIAA.