INTERACTIVE ORBITAL PROXIMITY OPERATIONS PLANNING SYSTEM

Arthur J. Grunwald and Stephen R. Ellis
NASA Ames Research Center, Moffett Field, California

ABSTRACT

An interactive, graphical proximity operations planning system has been developed which allows on-site design of efficient, complex, multiburn maneuvers in the dynamic multispacecraft environment about the space station. Maneuvering takes place in, as well as out of, the orbital plane. The difficulty in planning such missions results from the unusual and counterintuitive character of relative orbital motion trajectories and complex operational constraints, which are both time-varying and highly dependent on the mission scenario. This difficulty is greatly overcome by visualizing the relative trajectories and the relevant constraints in an easily interpretable, graphical format, which provides the operator with immediate feedback on design actions. The display shows a perspective bird's-eye view of the space station and co-orbiting spacecraft on the background of the station's orbital plane. The operator has control over two modes of operation: (1) a viewing system mode, which enables him or her to "explore" the spatial situation about the space station and thus choose and frame in on areas of interest; and (2) a trajectory design mode, which allows the interactive "editing" of a series of way-points and maneuvering burns to obtain a trajectory which complies with all operational constraints. Through a graphical interactive process, the operator will continue to modify the trajectory design until all operational constraints are met. The effectiveness of this display format in complex trajectory design is presently being evaluated in an ongoing experimental program.

INTRODUCTION

The future space station environment will include a variety of spacecraft co-orbiting with the space station in close vicinity. Mostly, these spacecraft will be "parked" in a stable location with respect to space station, i.e., they will be on the same circular orbit. However, some missions will require repositioning or transfers to and from these spacecraft. In these cases complex types of maneuvers are anticipated which involve a variety of spacecraft which are not necessarily located at stable locations and thus have relative motion between each other.

The multivehicle environment poses new requirements which do not exist in conventional missions scenarios. The conventional scenarios involve proximity operations between only two vehicles. In these two-spacecraft missions, the scenario is in most cases optimized and precomputed in advance, and executed at the time of the actual mission. However, since the set of possible scenarios in a multivehicle environment is virtually unlimited, the future space station environment will create scenarios which might not have been precomputed and will have to be planned and executed on site. This will require an on-site planning tool which allows, through a fast interactive process, the creation of a fuel-efficient maneuver which meets all constraints set by safety rules.

The difficulties encountered in planning and carrying out orbital maneuvers originate from several causes. The first is the counterintuitive character of orbital motions as experienced in a relative reference frame. The orbital motions are expressed in a coordinate frame attached to the space station and represent relative rather than absolute motions. It would be intuitively assumed that a thrust in "forward" direction, i.e., in the direction of the orbital velocity vector, would result in a straight-forward motion. However, after several minutes, orbital mechanics forces will dominate the motion pattern and move the spacecraft "upwards," i.e., to a higher orbit. This will result in a backwards relative motion, since objects in a higher orbit move slower. Thus, a forward thrust has an effect opposite from that intended.

A second cause of the difficulty is the different and unconventional way in which orbital maneuvering control forces are applied. In atmospheric flight, control forces are applied continuously to correct for randomly appearing atmospheric disturbances, or to compensate for atmospheric drag. In contrast, spaceflight in the absence of atmospheric disturbances has a near-deterministic character. Therefore, spaceflight is mainly "unpowered" along a section of an orbit with certain characteristics. By applying relatively short impulse-type maneuvering forces at a given way-point, the characteristics of the orbit will be altered. After application of the maneuvering force, the spacecraft will coast along on the revised orbit until the next way-point is reached.

Third, multivehicle orbital missions are subject to stringent safety constraints, such as clearance from existing structures, allowable approach velocities, angles of departure and arrival, and maneuvering burn restrictions due to plume impingement. Design of a fuel-efficient trajectory which satisfies these constraints is a nontrivial task.

It is clear that visualization of the relative trajectories and control forces in an easily interpretable graphical format will greatly improve the feel for orbital motions and control forces and will provide direct feedback of the operator's control actions. Furthermore, visualization of the constraints in a symbolic graphical format will enable an interactive graphical trajectory design in which, in each iteration step, the design is modified until all constraints are satisfied.

DESCRIPTION OF THE TECHNIQUE

Purpose of Orbital Planning System

The purpose of the interactive orbital planning system is to enable the operator to design an efficient, complex, multiburn maneuver, subject to the stringent safety constraints of the future dense space station traffic environment, which enables a chaser to rendezvous with a target spacecraft in a given timespan. The constraints include clearances from structures, relative velocities between spacecraft, angles of departure and arrival, approach velocity, and plume impingement. Because of the complexity and counterintuitiveness of orbital motion, and the demands to satisfy strict safety rules and constraints, fuel-efficient trajectory design will be a complex and difficult task. The basic idea underlying the system is to present the maneuver, as well as the relevant constraints, in an easily interpretable graphical format. This format provides operators with immediate feedback on the results of design actions, and enables them to closely interact with the system. In an iterative process, operators will keep changing the design until all constraints are met. The methods for enabling interactive trajectory design and visualization of constraints are discussed in detail hereafter.

Illustrative Example of a Three-Burn Maneuver

An illustrative example of a three-burn maneuver is shown schematically in figure 1, showing the situation in the orbital plane. Trajectory design can be greatly simplified by expressing the positions and velocities of co-orbiting spacecraft relative to a space-station-based coordinate system. This system $x°y°z°$ has its origin at the the center of mass of the station and is oriented with the $x°oy°$ plane locally level with the surface of the Earth, with the $x°$-axis in the direction of the station's orbital velocity vector and the $z°$-axis pointing towards the center of the Earth. Thus, the $x°oz°$ plane constitutes the orbital plane. The section of the circular orbit s, followed by the center-of-mass of the space station is called the "V-bar," and the radial line r, moving outwards from the Earth center through the space station, is called the "R-bar." For the near environment of the space station, the V-bar can be considered to be straight and to coincide with the $x°$-axis, and the R-bar with the $z°$-axis.

The trajectory originates from relative position A at time $t = t_o$ and is composed of two way-points B and C, which specify the location in space station coordinates at which the chaser spacecraft will pass at a given time. At a way-point the orbital maneuvering system or other reaction control system can be activated, creating a thrust vector of given magnitude for a given duration, in a given direction in the orbital plane or out of the orbital plane. The duration of the burn is considered very short in comparison with the total duration of the mission. In the orbital dynamics computations this means that a maneuvering burn can be considered as a velocity impulse which alters the direction and magnitude of the instantaneous orbital velocity vector of the spacecraft.

Since the initial location A is not necessarily a stationary point, the magnitude and direction of the relative velocity of the chaser at point A is determined by the parameters of its orbit. If no maneuvering burn would be initiated at $t = t_o$, the chaser would continue to follow the relative trajectory 1, subject to the parameters of its original orbit (see dotted line in fig. 1). However, a maneuvering burn at $t = t_o$ will alter the original orbit such that the chaser will follow the relative trajectory 2, subject to the parameters of a new orbit.

In figure 1 $\underline{v}_1$ and $\underline{v}_2$ indicate the relative velocity vector of the chaser just before and after the maneuvering burn, respectively, where $\underline{v}_1$ and $\underline{v}_2$ are tangential to the relative trajectories 1 and 2, respectively. The vector difference between $\underline{v}_1$ and $\underline{v}_2$, $\underline{v}_a$, is the velocity change initiated by the burn, and corresponds with the direction and magnitude or duration at which the orbital maneuvering system is activated. Likewise, at way-point B the burn $\underline{v}_b$ alters the orbit to orbit 3.

Location C is the terminal way-point and is, in this case, the location where the target will arrive at $t = t_f$. Since the target has an orbit of its own, orbit 4, it will have a terminal velocity at $t = t_f$. The relative velocity between target and chaser is the vector difference between $\underline{v}_3$ and $\underline{v}_4$, $\underline{v}_c$. This vector determines the retroburn that is needed at the target location, in order to bring the relative velocity between chaser and target to the minimum required for the docking operation.

Inverse Method of Solving Orbital Motion

Interactive trajectory design demands that the operator is given free control over the positioning of way-points. However, the input variables of the commonly used equations of orbital motion, as given in reference 1 and derived from references 2-4, are the magnitude and direction of the burn at $t = t_o$, rather than the position of way-points. Therefore an "inverse method" is

required to compute the values of a burn necessary to arrive at a given way-point positioned by the operator. This method is outlined hereafter.

The equations in reference 1 show how the orbital parameters of a co-orbiting spacecraft can be computed from its momentary position and velocities, relative to the space station. Thus, for a given initial relative position A with $\underline{x}(t_o)$, and an initial relative velocity $\underline{v}(t_o)$, at time $t = t_o$, the relative position and velocities of a way-point at time $t = t_1$ can be computed. However, a maneuvering burn at $t = t_o$ will cause a change in the direction and magnitude of the relative velocity vector $\underline{v}(t_o)$. As a result, the position of the way-point at time t_1, $\underline{x}(t_1)$ will change as well.

Consider v_a and α_a to be the magnitude and direction of the velocity change due to the maneuvering burn. Then the relative position and velocity at $t = t_1$, $\underline{x}(t_1)$, will be a complex, nonlinear function of v_a and α_a. Consider now that the operator is given direct control over v_a and α_a by slaving these variables directly to the x and y motions of an input device such as a control stick or mouse. An input in either x or y direction will result in a complex nonlinear motion pattern of $\underline{x}(t_1)$. Furthermore, this motion pattern will change with the initial conditions. This arrangement is highly undesirable in an interactive trajectory design process in which the operator must have direct and unconstrained control over the positioning of way-points.

It is therefore essential to give the operator direct control over the position of way-points rather than over the magnitude and direction of the burn. The inverse method by which this is accomplished computes the magnitude and direction of the burn required to bring the spacecraft from initial location $\underline{x}(t_o)$ to the way-point $\underline{x}(t_1)$ at $t = t_1$.

A Newton-Raphson method has been employed to solve this inverse problem. The operator commands the position of a way-point by means of the x-y motions of the input device. The algorithm starts with an initial guess of v_a and α_a. These values yield a computed way-point which is usually different from the commanded one. At each program update the values of v_a and α_a are adjusted to bring the computed way-point closer to the commanded one. On the average about three to four iterations are required to bring the difference between the computed and commanded way-point effectively to zero. As the operator moves the commanded way-point around in the orbital plane, the algorithm "tracks" the commanded way-point by continuously making appropriate adjustments in v_a and α_a. As a result of this continuous adjustment, the deviation between commanded and computed way-point will remain relatively small and the Newton-Raphson scheme will operate close to the optimum. The advantage of the Newton-Raphson scheme is that convergence with this second-order technique is the best in the near vicinity of the optimum. Since the program update rate is about 15 Hz, convergence is very fast and the computed way-point is virtually indistinguishable from the commanded one.

The Active Way-Point Concept

Although a trajectory may be composed of several way -points, only one way-point at a time, the active way-point, is controlled by the operator. The active way-point should be clearly distinguishable from the other inactive points, by conspicuous marking, highlighting, or blinking. While the position and time of arrival of the active way-point can be varied, the position and time of arrival of all other way-points remains unchanged. However, variations in the active way-point

will cause changes in the trajectory sections and way-point maneuvering burns just preceding and just following the active way-point. The on-line solution of the inverse algorithm enables these changes to be visualized almost instantaneously and provides the operator with on-line feedback on the design actions.

Although impingement constraints and approach velocity limits exist for all way-points, it is useful to limit the computation and display of these constraints to the active way-point only. This arrangement simplifies and speeds up system update computations and minimizes the symbology shown on the display. The justification for this is that the operator's attention is mainly allocated to the active way-point and its near vicinity. In a subsequent design iteration, the operator may shift the activation to a different way-point and again verify whether all constraints are met.

Since impingement constraints and approach velocity limits mainly relate to the target craft, it is useful to visualize the position of the target on the target trajectory, corresponding to the time of arrival at the active way-point. Like the active way-point itself, this position should be clearly distinguishable from other points as well.

Way-Point Editing

The trajectory design process involves changes in existing way-points, addition of new points, or deletion of existing undesired points. An illustrative example of this way-point editing process is shown in figure 2. In the program the way-points are managed by a way-point stack, which includes an up-to-date sequential list of the position $\underline{x}$, the time of arrival t, and the relative velocity $\underline{v}$ just after initiating the burn, of all way-points.

Figure 2a shows two way-points, the initial point $\underline{x}_0$ and the terminal point $\underline{x}_1$. The initial way-point is defined by the initial conditions of the situation and cannot be activated or changed by the operator. The terminal way-point $\underline{x}_1$ is thus the the active way-point which can be changed. The corresponding way-point stack is shown on the right. The active way-point box is drawn in bold. The relative velocity stack shows only the velocity $\underline{v}_0$, which is the required relative velocity just after the burn at way-point 0, computed by the inverse algorithm, to reach point $\underline{x}_1$ at time t_1.

Figure 2b shows the addition of a new way-point. This point is added half-way on the trajectory section just preceding the active way-point. Thus its time of arrival is chosen to be $t = 0.5(t_i + t_{i-1})$, where i in this case is 1 and relates to the stack before modification. The new position, $\underline{x}_1$ and relative velocity, $\underline{v}_1$ are computed by the "forward" equations given in reference 1, by computing the orbital position at the new time t, using the existing orbital parameters previously computed with $\underline{x}_0$, $\underline{v}_0$, and t_0. The newly computed way-point position, time and relative velocity are inserted between points 0 and 1 of the stack before modification and the new way-point is chosen to be the active one. The dotted lines in figure 2 indicate variables which are transferred without modification and the encircled variables are the newly computed ones. It is important to note that since the relative velocities $\underline{v}_0$ and $\underline{v}_1$ are matched to the required way-points $\underline{x}_1$ and $\underline{x}_2$, respectively, the inverse algorithm does not need to make any adjustments.

Figure 2c shows the results of changes in the newly created way-point on the way-point stack. Since $\underline{x}_1$ and t_1 are varied, the relative velocity at way-point 0, $\underline{v}_0$ will be readjusted by the inverse algorithm and likewise the relative velocity $\underline{v}_1$.

Figure 2d shows the creation of an additional new way-point. Since the active way-point prior to the addition was point 1, the new point is added half-way between point 0 and 1 and its position and relative velocity are computed with the forward method. The new values are inserted between points 0 and 1 of the stack before modification and the new way-point is again set to be the active one.

In figure 2e way-point 2 is activated. Apart from the shift in active way-point, the stack remains unchanged. The dotted line shows the the direct-path section between point 1 and point 3 without the intermediate burn at point 2. Deletion of way-point 2 will remove this point from the stack, and after that close the gap (fig. 2f). However $\underline{v}_1$ has to be readjusted to fit the new direct-path section. Starting from the old incorrect value of $\underline{v}_1$, the adjustment is made iteratively and on-line by the inverse algorithm.

Operational Constraints

The multispacecraft environment will require strict safety rules regarding the clearance from existing structures. Thus, spatial "envelopes" can be defined through which the spacecraft is not allowed to pass. These spatial constraints can be visualized on the display. The operator must be able to make a clear judgment whether the planned trajectory clears the spatial constraint, or, he or she must be able to decide whether to avoid the constraint through an in-plane or an out-of-plane maneuver. However, the operator is not always able to make these judgments on the basis of one perspective aerial view or one perspective projection. In this research a graphical enhancement is used in which the spatial constraint is unambiguously presented on a time-axis display format. This format and its advantages are discussed later.

Restrictions on angles of departure and arrival may originate from structural constraints at the departure gate, or the orientation of the docking gate or grapple device at the target craft. Limits for the allowable angles of departure or arrival can be visualized on the display. In addition, the terminal approach velocity at the target might be limited by the characteristics of the grapple mechanism or the docking procedure. Limits for the allowable terminal approach velocity can be visualized as well.

Way-point maneuvering burns are subject to plume impingement constraints. Hot exhaust gases of the orbital maneuvering systems may damage the reflecting surfaces of sensitive optical equipment such as telescopes, infrared sensors, or solar panels, or may cause an undesired transfer of momentum. Maneuvering burns towards these pieces of equipment are restricted in direction and magnitude. Limits for the allowable direction and magnitude are a function of the distance to the equipment and plume characteristics. These limits can be visualized on the display.

Flight safety requires that the relative velocity between spacecraft is subject to approach velocity limits. In conventional docking procedures this limit was proportional to the range (refs. 5-7). A commonly used rule of thumb is to limit the relative approach velocity to 0.1% of the range per second. This conventional rule is quite conservative and originates from visual procedures in which large safety margins are taken into account to correct for human or system errors. Although the future traffic environment will be more complex, and will therefore demand larger safety margins, more advanced and reliable measurement and control systems will somewhat relax these demands. The effect of these developments on the allowable approach velocity limits is at present difficult to predict and so is the margin for human error to be taken into account.

In this study, the relative approach velocity is defined as the component of the relative approach velocity vector between the two spacecraft along their mutual line of sight. The limit on this relative approach velocity is a function of the range between the spacecraft. This function will depend on the environment, the task, and the reliability of measurement and control equipment, and cannot be determined at this stage. In this study a simple proportional relation has been chosen. The approach velocity limit is visualized on the display as a circle indicating the minimum range between the two spacecraft allowed for the present approach velocity. If the target craft appears within this circle, the approach velocity limit has been violated.

DESCRIPTION OF THE DISPLAY

Graphics System and Layout of the Display Area

The system has been implemented on a Silicon Graphics IRIS 2400 Turbo Graphics Workstation with 24 bitplanes of display memory and with a 19-inch, full-color display monitor with a display resolution of 1024 by 767 pixels. The program is named "NAVIE," which is the Hebrew word for prophet, after the prophet Elijah, who was characterized by providing trustworthy future information. Operator interaction with the system is through a two-axis, three-button mouse.

The layout of the display area is shown in figure 3. The display area has been divided into four viewports. The main area 1 is 750 by 750 pixels and areas 2,3, and 4 are 230 by 230 pixels each. Viewports 1, 3, and 4 provide information about the spatial situation about the space station, trajectories, constraints, and orbital maneuvering fuel use; and viewport 2 includes an eight-button function control panel.

Description of Program Control Modes

The program operates in two modes. The first one, the viewing system mode, relates to the main display, which shows a perspective view of the space station and its surroundings on the background of the station's orbital plane. In the viewing system mode, the operator is able to "explore" the spatial situation about the space station and thus choose a viewpoint location and viewing direction which focuses and "frames in" on the momentary area of interest. The second mode is the trajectory design mode, in which way-points are selected, moved, added, and deleted in order to obtain a multiburn trajectory which complies with the given set of constraints.

Viewing System Mode

The geometry of the viewing situation is shown in figure 4. The space-station-based coordinate system is $x^{\circ}y^{\circ}z^{\circ}$ with the x°-axis coinciding with the orbital velocity vector, and $x^{\circ}oz^{\circ}$ is the orbital plane. Figure 4 shows the orientation of the viewing system $x^{e}y^{e}z^{e}$ relative to the space station system. The viewing system has its origin at point A, the x^{e}-axis coincides with the viewing direction and the image plane is perpendicular to the x^{e}-axis with the screen axes y^{s} and z^{s} parallel to y^{e} and z^{e}. Point B indicates the intersection of the viewing axis with the orbital plane. Although the viewing system position, point A, and the angular orientation are defined by three

displacements and three angles, which can be all controlled independently, it is useful to constrain the motion to the following three types.

"Tethered" motion- In the first type of motion, the viewing system tethers about point B, which is kept fixed on the orbital grid, while the distance d between points A and B, which is the viewing range to point B, is kept constant. The tethered motion is controlled by the angles ψ and θ. The viewing axis x^e and the axis y^e are located at all times in the plane P which passes through the point B and rotates about the line CC', which is parallel to the x°-axis, the V-bar. The line BE is also located in the plane P and perpendicular to the line CC'. ψ is the angle between the y°-axis and the line BE, and θ is the the angle between BE and the x^e-axis. Thus, the angles ψ and θ control the obliquity of viewing along the orbital plane in the z° and x° direction, respectively. This tethered type of motion is very useful for the following reasons. (1) While the area of interest remains in the center of the display, it allows one to "explore" other possible areas of interest by changing the angles ψ and θ. (2) The line CC' will appear on the screen at all times as a horizontal line through the center of the display and represents a line parallel to the V-bar. Thus, while the viewing direction may change, the direction of the V-bar is at all times recognizable as the horizontal line, passing through the center of the display.

Translational motion- The second type of motion relates to the position of point B in the orbital plane. Here the $x^\circ z^\circ$ coordinates of point B are varied, while ψ, θ, and d are kept constant. This translational type of motion enables the operator to move areas of interest to the center of the display.

Ranging motion- In the third type of motion, all parameters are kept constant except for the range d. This ranging type of motion is useful after areas of interest are located and brought into the center of the display. "Ranging-in" on the area of interest allows this area to be studied in more detail.

In the viewing system mode the operator has one-button control over the three types of motion and can "toggle" in a closed sequence from tethered motion to translational motion to ranging motion and back to tethered motion. The one-button control is useful since viewing system operations are naturally performed in a sequence of three steps, where in the first step areas of interest are searched for, in the second step the area localized during the search is moved to the center of the display, and in the third step the area is ranged in on to obtain the required level of detail.

Trajectory Design Mode

In the trajectory design mode, the operator has control over the selection, positioning, time of arrival, addition, and deletion of the way-points which determine the trajectory. Two submodes exist: the in-plane design mode and the out-of-plane design mode. In the in-plane mode the mouse controls the $x^\circ z^\circ$ position of way-points, while the out-of-plane position y° remains unchanged, whereas in the out-of-plane mode the opposite is the case.

The design process starts with an initial configuration of way-points. Usually there are initially two way-points, as in the way-point editing example. The terminal point $\underline{x}_1$ is the active way-point. Time of arrival at this active way-point is set to an initial value within the allowable time span of the mission. The operator has the option to increase or decrease the time of arrival at any

active way-point. The time of arrival at the terminal way-point is limited to the time span of the mission, and the one of an intermediate way-point by the time span set by the neighboring points.

As outlined previously, a convention is chosen in which a new way-point is added half-way on the time scale, on the trajectory section preceding the active way-point. The newly added way-point becomes the active one and can be moved to any desired location and its time of arrival can be set to any value within the time span determined by the neighboring way-points. However, in some cases, it is useful to "slide" the new way-point along the trajectory section connecting its neighboring way-points. The position on this trajectory section is then determined by its time of arrival only. In this mode the "locked-on-trajectory" mode, the time of arrival is slaved to the y-motions of the mouse.

The locked-on-trajectory mode is particularly useful for checking whether operational constraints between the spacecraft and the target, or other nonstationary spacecraft, are being violated. As the operator slides the way-point along the trajectory, the corresponding target position slides along the target trace as well; conflicting situations, such as a too close flyby, will be recognized immediately.

Geometrical Enhancements; the "Time-Axis" Format

The purpose of these enhancements is to resolve ambiguities in the spatial situation by processing the spatial information and presenting it in a different format. One such format is the time-axis display which provides unambiguous qualitative and quantitative information about the out-of-plane situation and the spatial constraints.

The basic idea of the time-axis format is demonstrated in figures 5a-c. From the perspective view of figure 5a alone, it cannot be clearly determined whether the spatial constraint is violated or how the trajectory should be planned to avoid it. The view along the z°-axis in figure 5b is even less clear, because of the curved character of the trajectory. In the time-axis format of figure 5c, the out-of-plane deviation is plotted as a function of the traveled time along the path. The spatial constraints are visualized as follows. At each point on the traveled time axis, at the corresponding location on the trajectory, a line is placed perpendicular to the orbital plane. Sections of this line which are within these constraints are identified and plotted on the time-axis display of figure 5c as a set of vertical bars. Where the trajectory curve passes through these bars, the spatial constraints have been violated. Reshaping of the in-plane trajectory will alter the size and location of the constraint bars on the time-axis display. From the display it can be clearly determined whether the constraint should be avoided through an in-plane or an out-of-plane maneuver.

The format of the time-axis display used in the program is shown in figure 6. The time-axis is marked in quarters of an orbit. The shaded areas represent the nighttime section of the orbit. Both the target and the chaser trajectories are shown. It should be noted however, that although the chaser and target share the same time axis, they relate to different spatial trajectories. Therefore, the spatial constraint bars relate to the chaser trajectory only.

Symbolic Enhancements

Visualization of departure constraints- Procedures at the departure gate might constrain the relative angle of departure and the magnitude of the departure burn. The in-plane constraints at the departure gate are illustrated in figure 7. The size of the burn vector is made proportional to the burn magnitude, with a scale factor of 500-m length per 1-m/sec burn, on an orbital grid with lines spaced 200 m apart. The departure constraints are satisfied if the burn vector is within the solid "bracketed" arc. This arc is specified by the arc center angle γ_0, the arc aperture γ, and the arc radius ε. Note that maneuvering burns are expressed in terms of a velocity change rather than of a thrust force. The actual duration and thrust force of the burn depends on the spacecraft mass and the thruster characteristics.

In order to keep the display free from unnecessary symbology, it is useful to present the constraint only when it is close to being violated. If the burn vector is within the area enclosed by the dotted line in figure 7, the constraint is not drawn. The radius of the dotted arc is 80% of ε, and the aperture angle is 10° smaller than γ.

It should be noted that the situation in figure 7 relates to a stationary departure gate. The spacecraft trajectory in this case is aligned with the burn vector. For a departure gate which moves with respect to the space station system, this will not be the case. In this case the burn vector will signify the relative direction of departure with respect to the moving gate, rather than with respect to the space station. But this vector is subject to the departure constraints and not the velocity vector of the trajectory, which is relative to the space station. Therefore, the symbology is valid for departure from a stationary as well as a nonstationary gate.

The out-of-plane constraint at the departure gate is illustrated in figure 6. The initial out-of-plane component of the burn vector has to be within the impingement constraint brackets. The out-of-plane burn scale factor is 500-m length per 1-m/sec burn. If the burn magnitude is less than 80% of the allowed maximum value, the constraint is not drawn.

Visualization of arrival constraints- The arrival procedures constrain the angle and magnitude of the terminal velocity vector relative to the arrival gate. The in-plane constraints at the arrival gate are visualized in figure 8. The scale factor f r the relative terminal velocity vector is 500-m length per 1-m/sec terminal velocity. The arrival constraints are satisfied if this vector is within the solid arrival arc. This arc is specified by the arc center angle δ_0, the arc aperture δ, and the arc radius η. The arrival arc is visualized at all times.

The out-of-plane limits on the terminal approach velocity are depicted in figure 6. The approach velocity has to be within the constraint brackets. If the velocity is less than 80% of the allowed maximum value, the constraint is not drawn.

Visualization of plume impingement constraints- Plume impingement constraints limit the magnitude and direction of maneuvering burns. The in-plane impingement constraints of a burn given at a way-point towards the target are illustrated in figure 9. The burn-vector symbol, whose size is proportional to the magnitude of the burn, is not allowed to cross the bracketed impingement constraint arc with aperture β and radius σ. The variables β and σ are a function of the distance between way-point and target $|\Delta\underline{X}| = |\underline{X}_T \sim \underline{X}|$, whose function depends on the characteristics of plume and target. In this example, B is chosen to be constant and σ propor-

tional to $|\Delta\underline{x}|$. If the burn vector does not cross the dotted bracketed arc, the constraint is not drawn. The radius of the dotted arc is again 80% of σ and the aperture angle is 10° larger than β.

Visualization of the approach velocity constraint- The method of visualizing the relative approach velocity limit is shown in figure 10. The relative approach velocity of the chaser towards the target is given by the vector $\Delta\underline{v} = \underline{v} - \underline{v}_T$. The line-of-sight vector of the chaser towards the target is $\Delta\underline{x} = \underline{x}_T - \underline{x}$. The relative approach velocity vector $\underline{v}_r$ is the projection of $\Delta\underline{v}$ on $\Delta\underline{x}$ and is given by

$$\underline{v}_r = (\Delta\underline{v}^T\Delta\underline{x})\Delta\underline{x}/|\Delta\underline{x}|^2 \quad (1)$$

where T denotes the transpose, or inner product. The limit on $|\underline{v}_r|$ is a function of the distance between chaser and target $|\Delta\underline{x}|$. In this example, a simple proportional relationship has been chosen. Thus, for a given approach velocity $|\underline{v}_r|$, the allowable range ρ can be computed and visualized by a circle centered about the chaser's position. The approach velocity constraint is violated when the target is located within this circle. The circle is visualized when ρ is greater than 80% of $|\Delta\underline{x}|$.

Orbital fuel use- The orbital fuel use is displayed in viewport 4. The orbital fuel is expressed in total m/sec velocity change rather than kg fuel mass. The actually spent fuel mass depends on the spacecraft and the thruster characteristics and will be proportional to the total velocity change. A fuel dial is shown which indicates the percentage of fuel remaining from the total amount allowed for the mission. The remaining fuel is indicated by a yellow sector, and fuel use in excess of the allowed amount is indicated by this sector turning red. In addition to the fuel dial, the percentage of fuel left and total fuel use are displayed numerically.

Trajectory time markers- Along the chaser and the target trajectories, time markers are placed at regular intervals. The time marker is a small bar, perpendicular to the trajectory, provided with a number which indicates the time in minutes after starting the maneuver. Special care is given to the automatic repositioning of the numericals after a viewing system change. The numericals are placed such that they do not "clutter" the trajectory and clearly point to the corresponding time marker.

Computational Enhancements

Computation of the relative trajectories is a time-consuming process, which, if done at each program update, will result in an unacceptable low update rate, jerky motions, and poor control over the positioning of a way-point. This can be prevented by disabling the trajectory computations and starting them only after the operator has completed the positioning of a way-point. At each program update interval, the x and y output values of the mouse are compared with the values from the previous step. If no change has taken place, a timer is initiated. The trajectory computations are started 0.3 sec after initiating the timer. After the trajectory is computed, the computed values are stored and displayed and no further computations will take place until the next change in way-point position. The 0.3 sec delay is essential for assuring that the operator has completed the positioning process. Often, small corrections are made after the way-point has been moved the first time. Experience has shown that, in most cases, no more changes are made after a 0.3 sec delay. Sometimes subsequent changes are made after the operator has reviewed the position. These

changes are seldom made earlier than 0.5 sec after the last change and this is after the trajectory has been recomputed.

It should be noted that although the trajectory computations are subject to delay, this is not the case with the computation of variables which relate to the way-points themselves, such as maneuvering burn vectors, relative velocity vectors, and operational constraints. The computation of these variables is less time-consuming and is done at each program update interval. Continuous update of these variables is essential in order to give the operator immediate feedback of the effect of a certain design action on maneuvering burns or approach velocities.

DISCUSSION

The proposed interactive orbital planning system should be seen as a preliminary step in determining the display format which will be useful in the dense space station environment. The examples shown here deal with the most general situation, which involves departures from, and arrival at, nonstationary locations. However, most of the co-orbiting spacecraft are likely to be "parked" on the V-bar, and thus at stationary positions. Missions with spacecraft at nonstationary positions and substantial out-of-plane motion thus represent a worst-case situation, and are chosen here to demonstrate the capabilities of interactive graphical trajectory design, rather than representing the common type of maneuver to be executed at the station.

Likewise, it is hard to predict whether the constraints used here will be relevant and realistic in the future space station environment. They predict in a broad sense the type of restrictions which are expected in the multivehicle environment, e.g., limitations on approach rates, plume impingement, and clearance from structures. It is also likely that the future environment will pose different constraints, which might originate from the specific character of a mission, like a specific scenario in which a telescope or manufacturing platform is approached and serviced.

A further restriction of the display relates to the way the orbital maneuvering system is activated. Only pure impulse maneuvering burns are considered, in which the duration of the burn is negligible with respect to the duration of the mission and in which these burns cause major changes in the relative trajectories. Station-keeping or fly-by missions, however, require a more sustained type of activation, such as periodic small burns with intervals of several seconds over a time span of several minutes. A more distributed way of activating the orbital maneuvering system can be introduced in which the operator has control over the frequency and time span of the activation. Ways should be found which enable this type of control to be activated and visualized.

A last restriction relates to the way the spatial trajectory is visualized. The perspective main view shows the projection of the actual trajectory on the orbital plane, rather than the trajectory itself. The reason for this is two-fold. The orbital trajectory, with its typical cycloidal shape, when shown without lines projected on the orbital reference plane is ambiguous and might seem to bend out of the orbital plane. This illusion results from the viewer's familiarity with objects such as a coil spring and has first been reported in reference 8. Therefore, the trajectory cannot be shown without its projection on the orbital plane. Second, the symbolic enhancements and burn vectors relate to the in-plane motion and match with the trajectory projection on the orbital plane. Thus, both the trajectory and its projection should actually be visualized. However, in a perspective plan view, i.e., viewed along the y°-axis, both the trajectory and its projection on the orbital plane will

show up as separate curves which might be highly confusing. Therefore a compromise has been sought, in which the projection is shown together with "pedestals" placed at the way-points orthogonal to the orbital plane, which mark the actual trajectory at the way-points.

In spite of these restrictions, the proposed display clearly demonstrates the usefulness of interactive graphical trajectory design. The use of the graphical, symbolical, and computational enhancements indicates the direction in which a solution for a multivehicle environment display should be sought. A still-unanswered question relates to the degree of automatization which should be introduced in the display. Parts of the mission could be performed through the use of optimization techniques, e.g., to find the fuel-optimal way-point which clears a spatial constraint in part of the mission, or to find a way-point which satisfies the terminal constraints. However, since the solution space of a complex situation is virtually infinite, it is yet doubtful whether this mission can be performed entirely automatically. It is therefore expected that frequently occurring routine operations, such as searching the local solution space for the optimal location of a way-point, might be handed over to an optimization scheme. These solutions can be reviewed by the operator, and manually changed if necessary.

In a presently ongoing experimental program, operators are carrying out a series of design missions which vary in complexity and constraints. In a tutorial session, the operators are first familiarized with the orbital motions, orbital control methods, operational constraints, and the system control functions of the viewing system motions and way-point editing process. Each operator action is time-marked and recorded. Statistics of the viewing system actions will show "preferred" viewing situations for each condition. Review of the trajectory design actions might identify the existence of heuristic design rules which might be utilized in automated design schemes.

REFERENCES

1. A.J. Grunwald and S.R. Ellis, "Interactive Orbital Proximity Operations Planning System," NASA TP-2839, 1988.

2. L.G. Taff, Celestial Mechanics, A Computational Guide for the Practitioner, John Wiley & Sons, 1985.

3. J. Kovalevsky, Introduction to Celestial Mechanics, Translated by Express Translation Service, Springer-Verlag, New York, 1984.

4. W.T. Thomson, Introduction to Space Dynamics, Dover, New York, 1986.

5. NASA Lyndon B. Johnson Space Center, "Flight Procedures Handbook," JSC-10566, Nov. 1982.

6. A.R. Brody, "Spacecraft Flight Simulation: A Human Factors Investigation into the Man-Machine Interface between an Astronaut and a Spacecraft Performing Docking Maneuvers and other Proximity Operations," M.Sc. Thesis, Massachusetts Institute of Technology, April 1987.

7. NASA Lyndon B. Johnson Space Center, "Rendezvous/Proximity Operations Workbook," RNDZ 2102, 1983.

8. S.R. Ellis and A.J. Grunwald, "A New Illusion of Projected Three-Dimensional Space," NASA TM 100006, 1987.

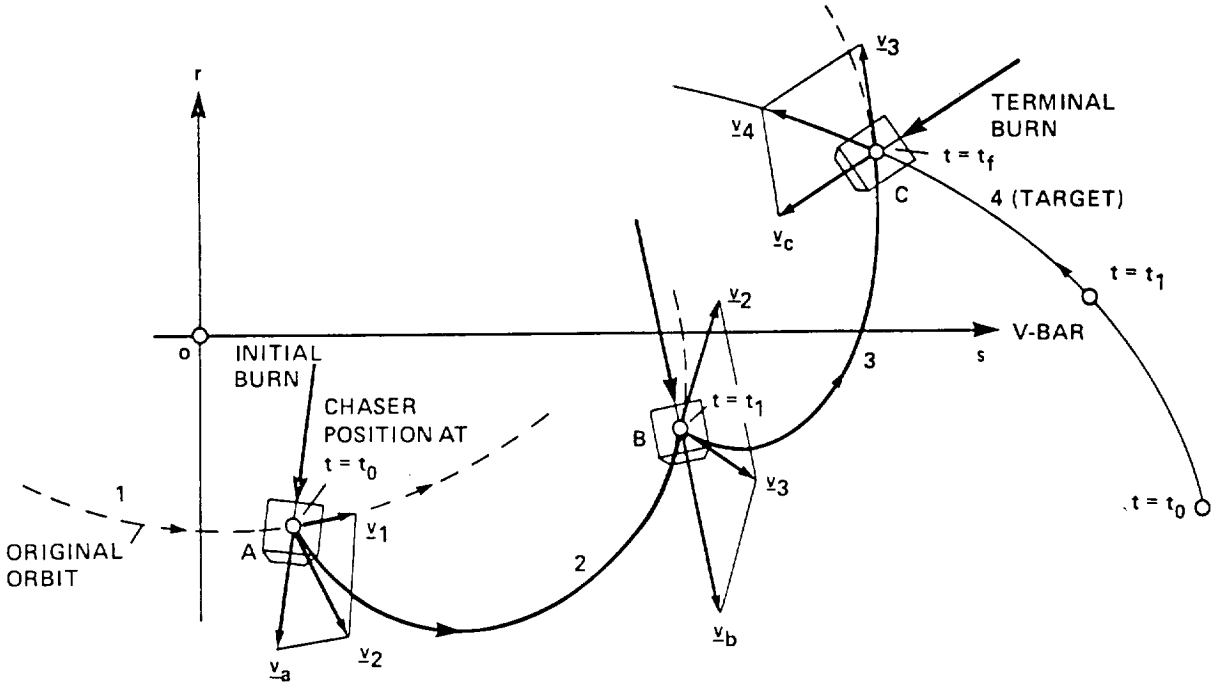

Figure 1.- Example of a three-burn maneuver.

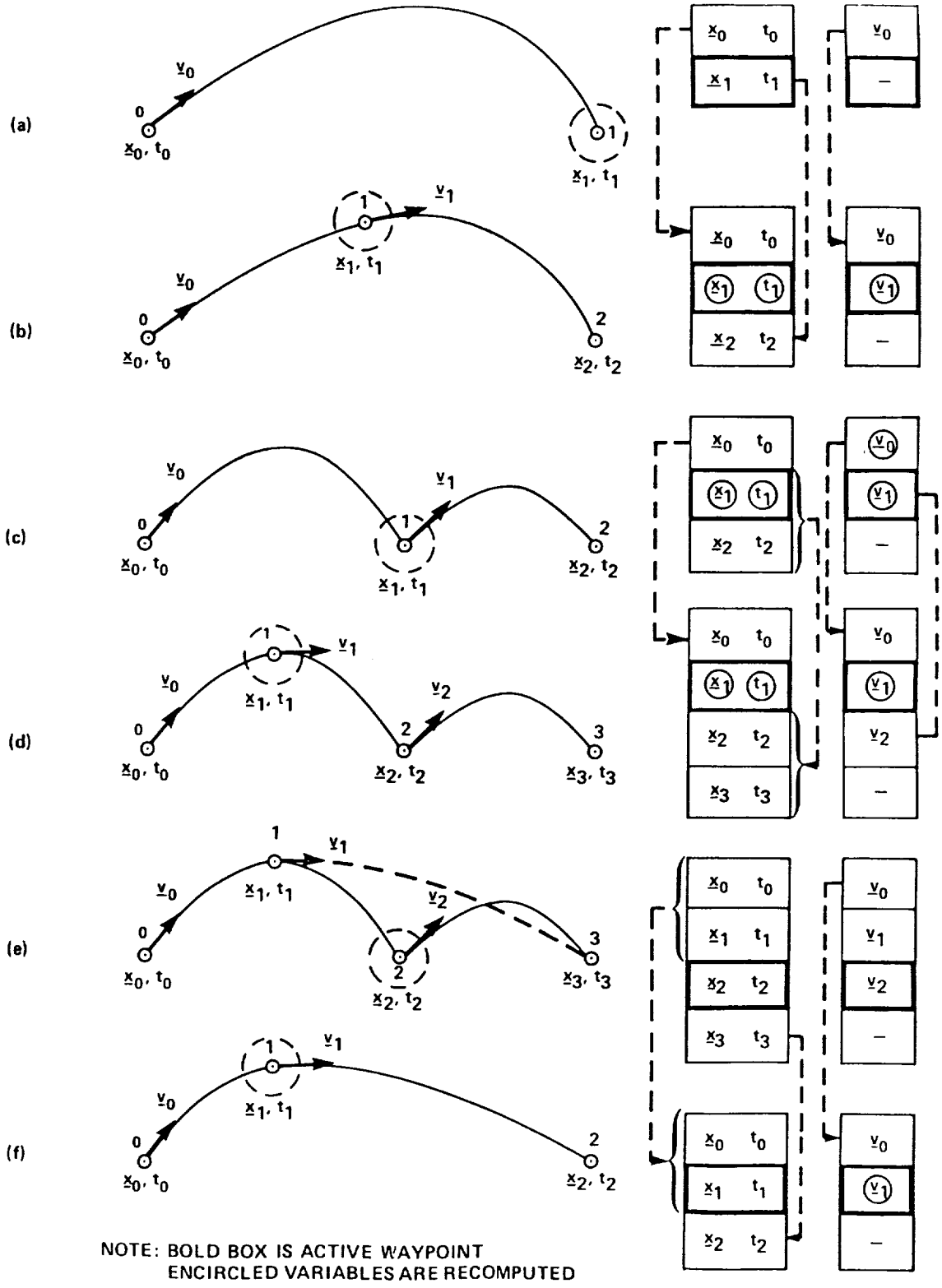

Figure 2.- Editing of way-points.

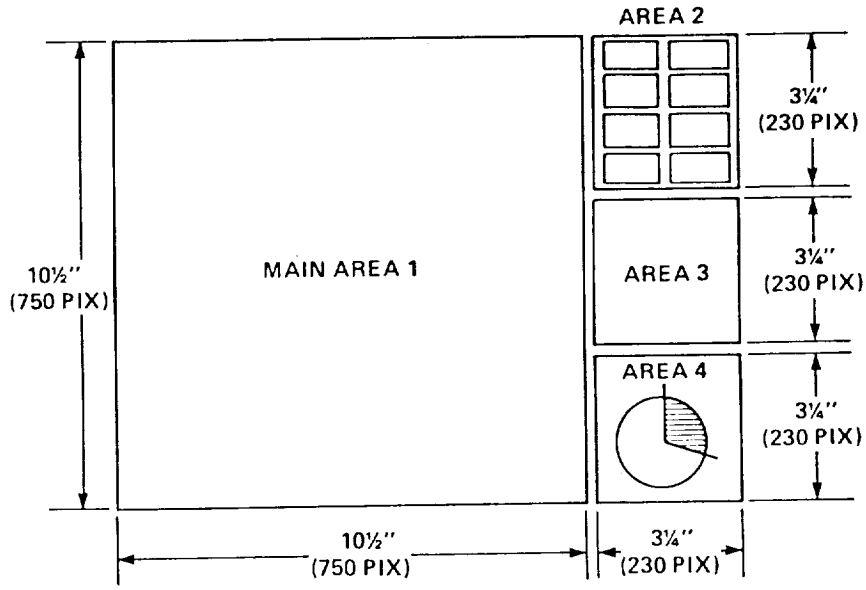

Figure 3.- Layout of the display area.

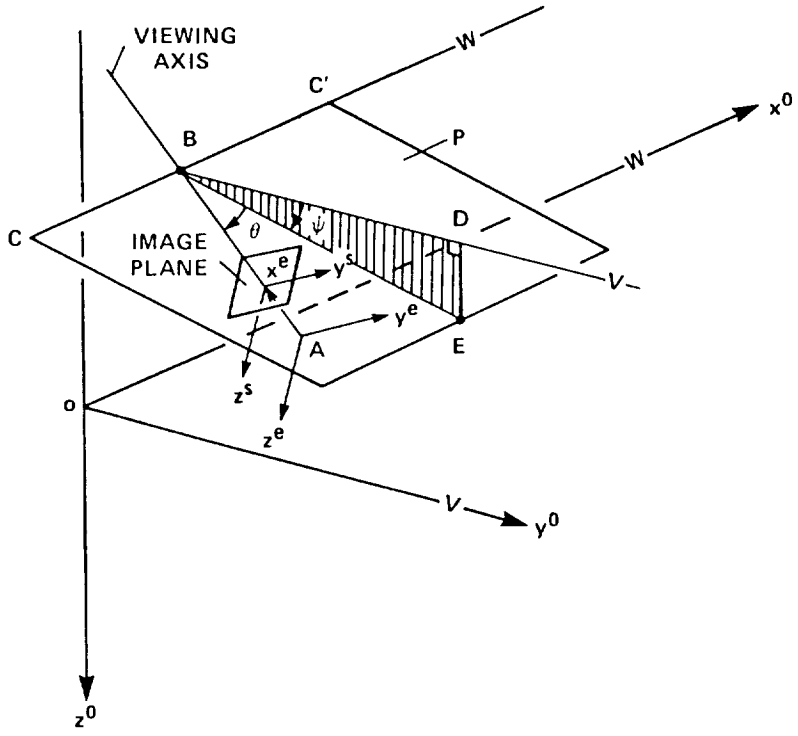

Figure 4.- Geometry of the viewing situation.

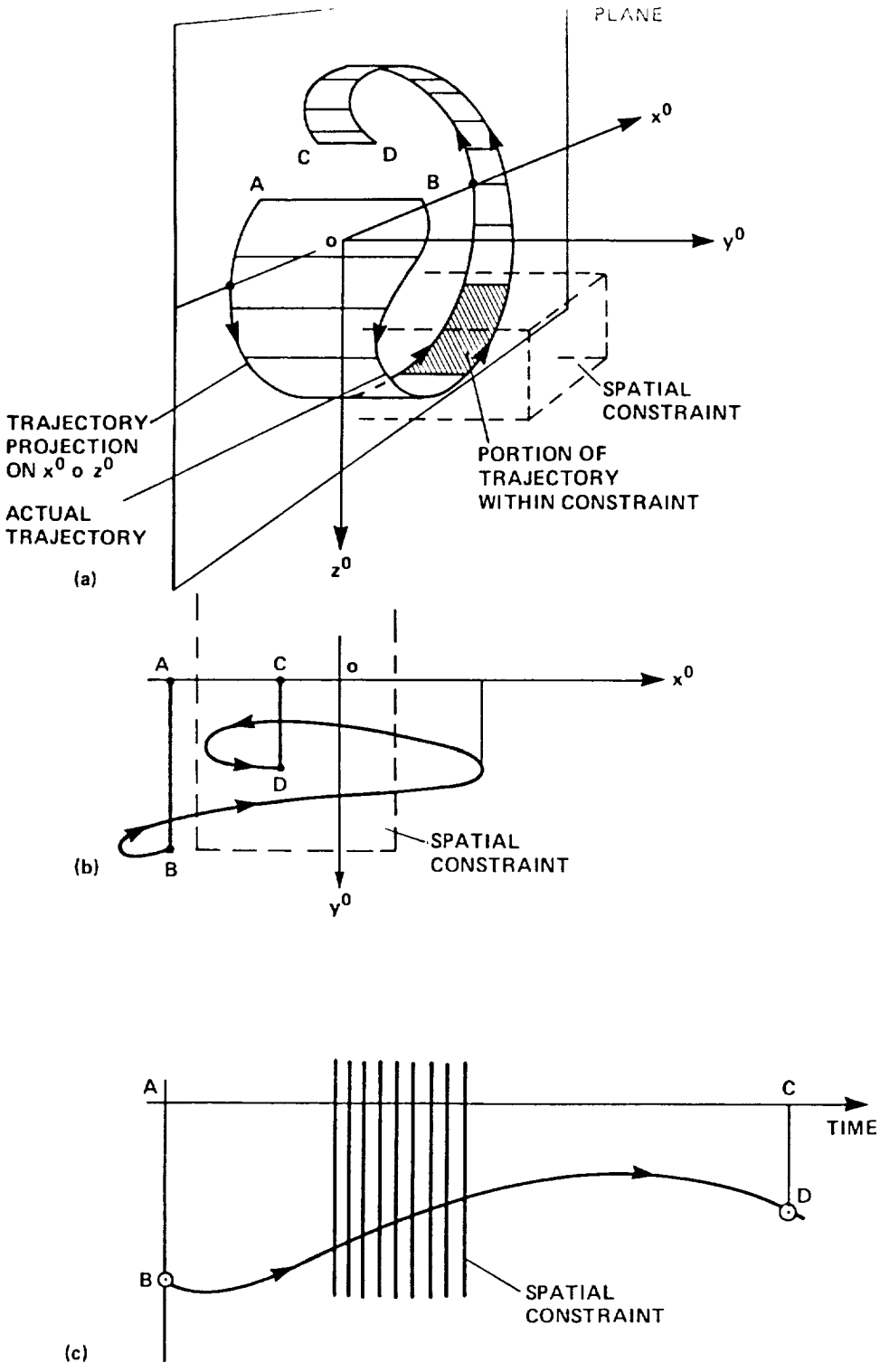

Figure 5.- Principle of the time-axis format. (a) Perspective view. (b) View along the z°-axis. (c) Time-axis format.

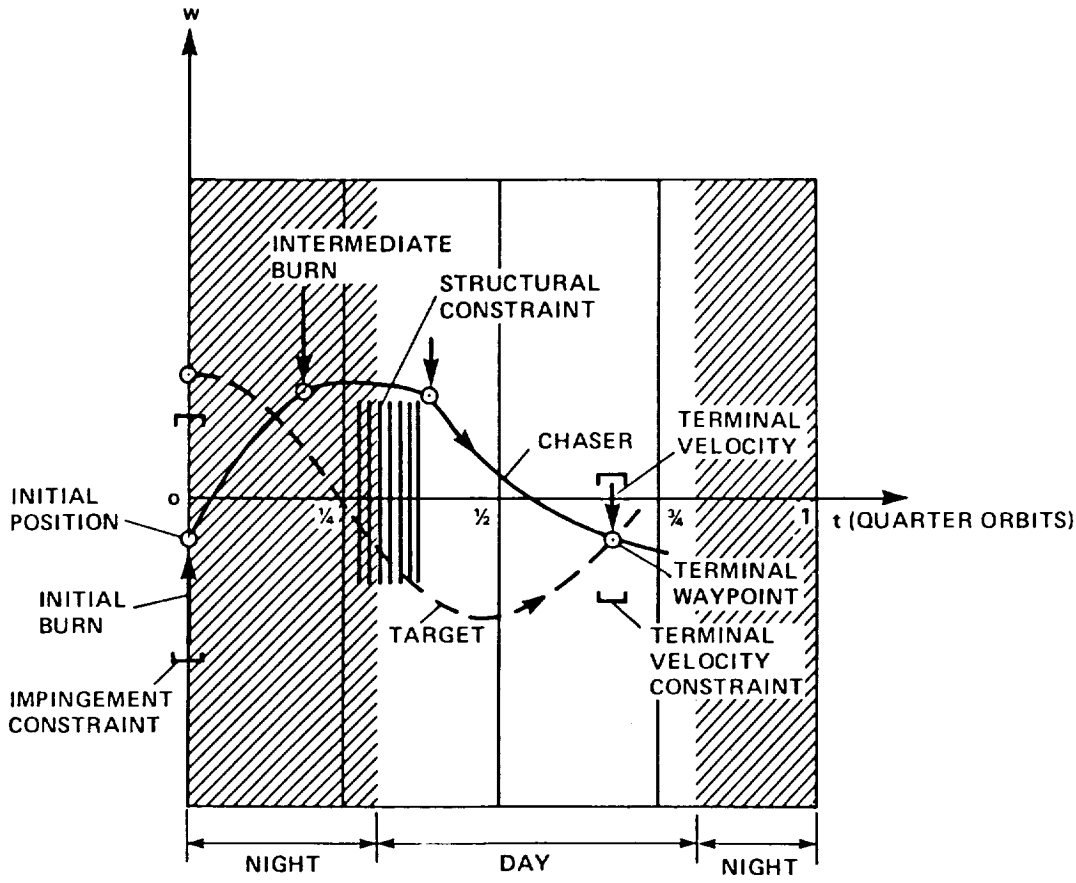

Figure 6.- Time-axis format with operational constraints and day/night indication.

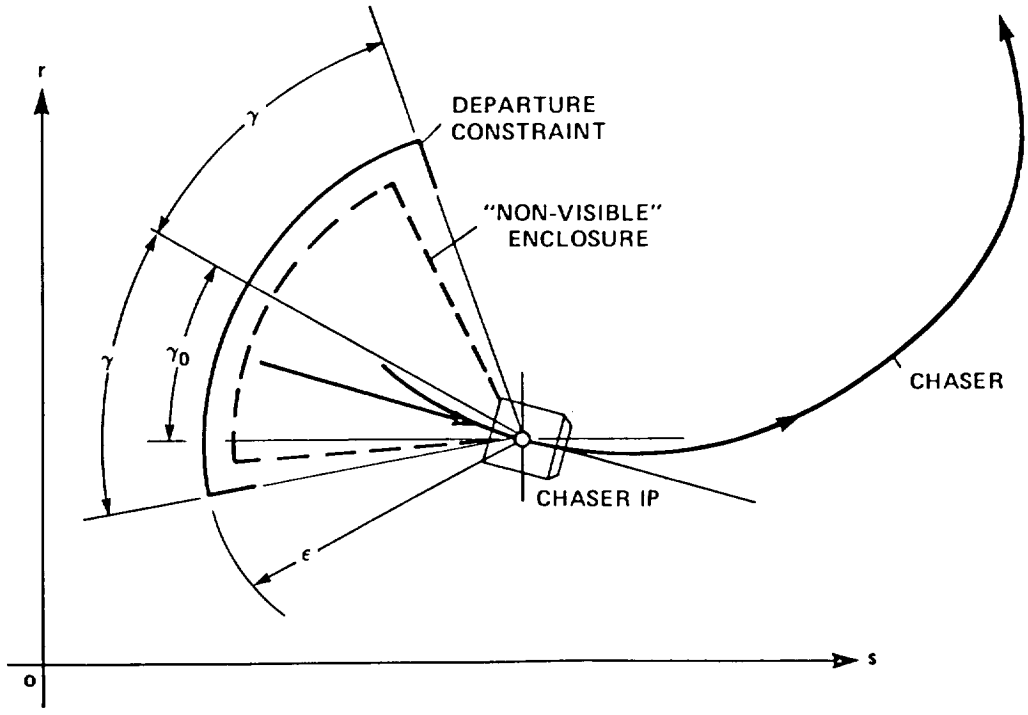

Figure 7.- Departure constraints.

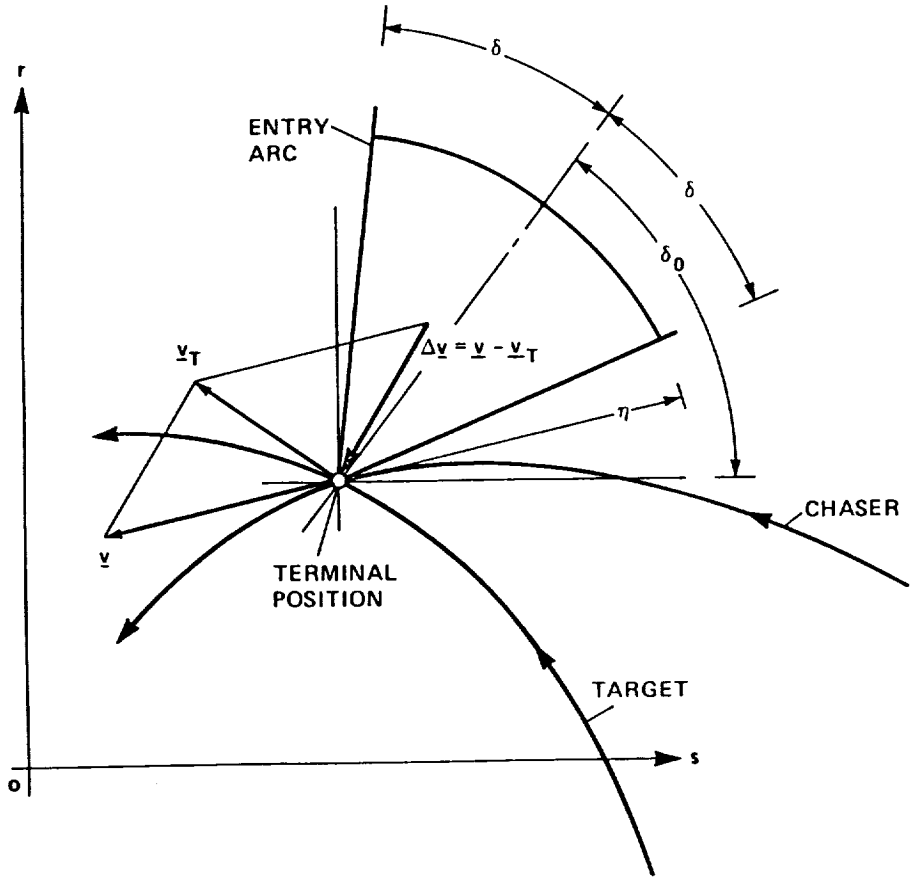

Figure 8.- Arrival constraints.

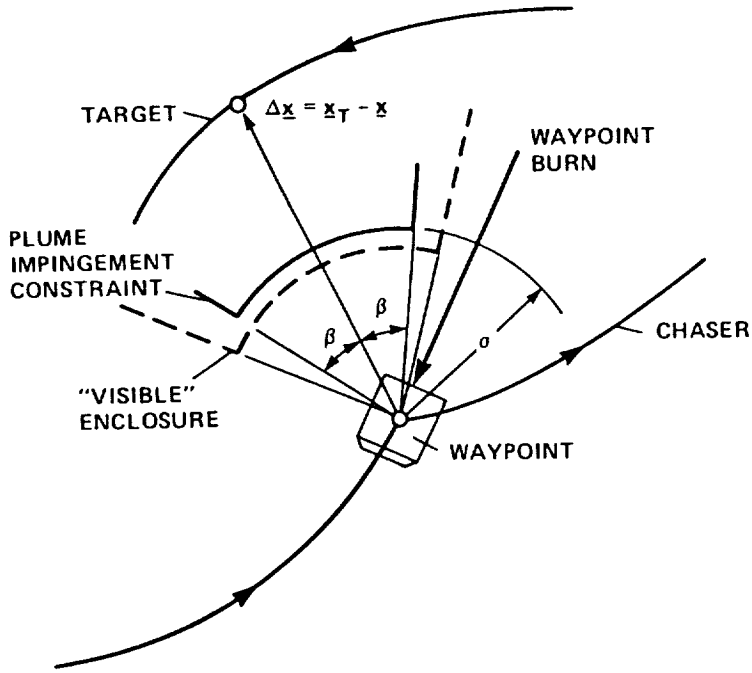

Figure 9.- Plume impingement constraints.

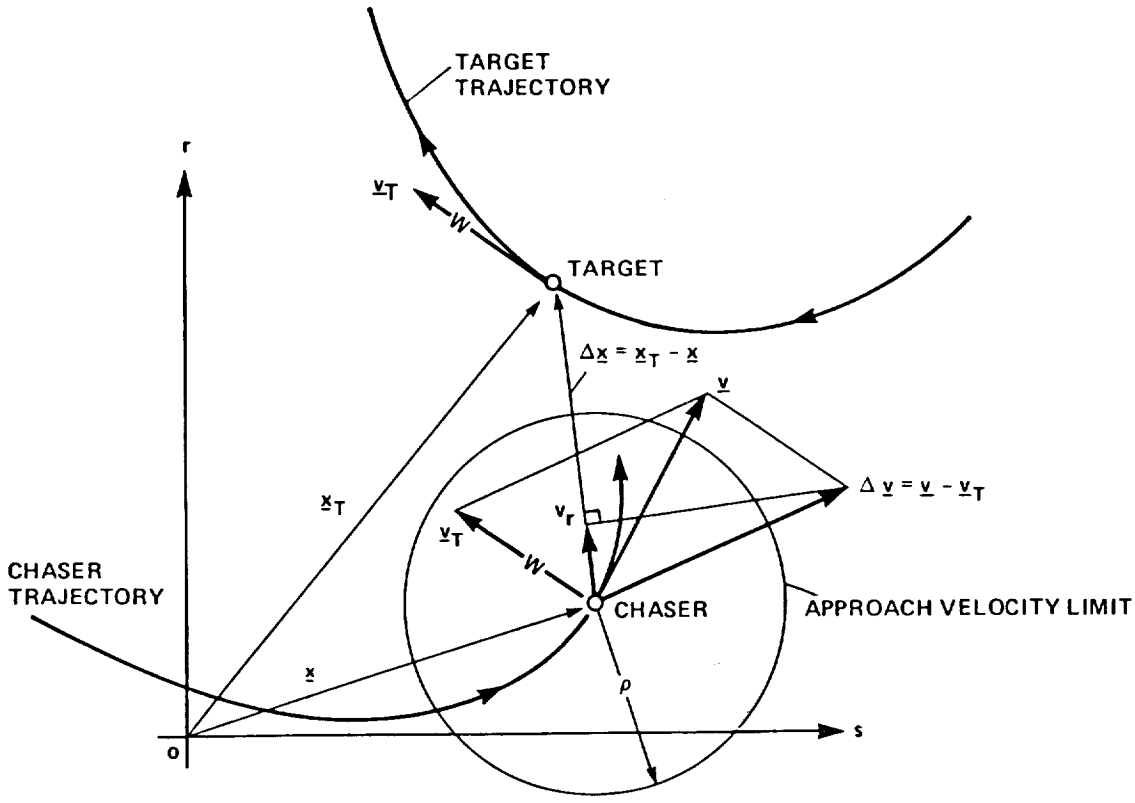

Figure 10.- Approach velocity limit circle.