



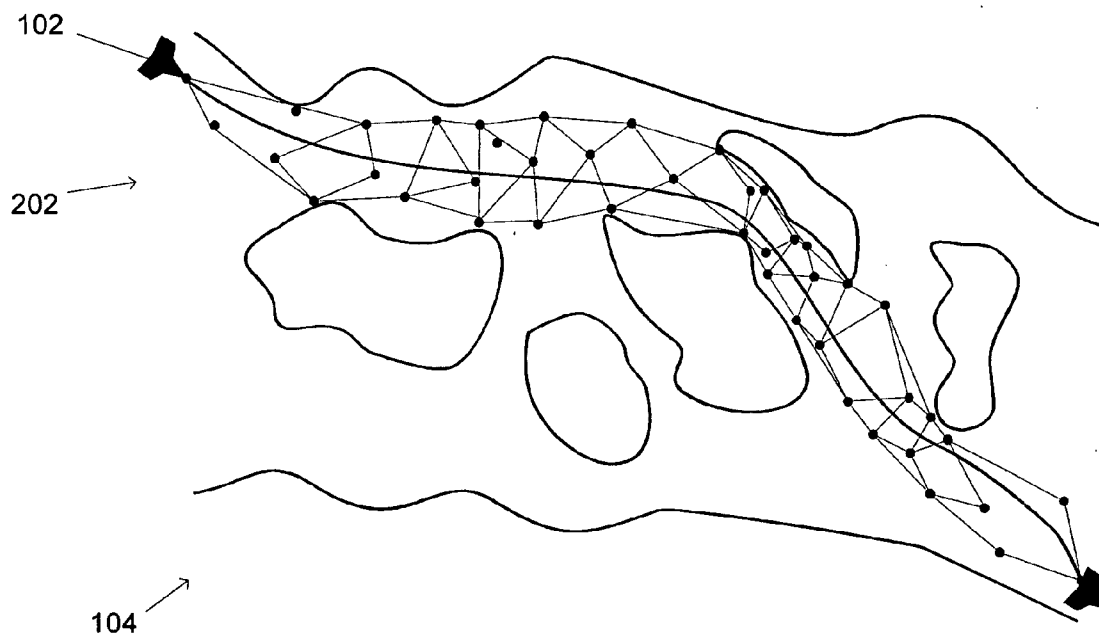
US 20070179685A1

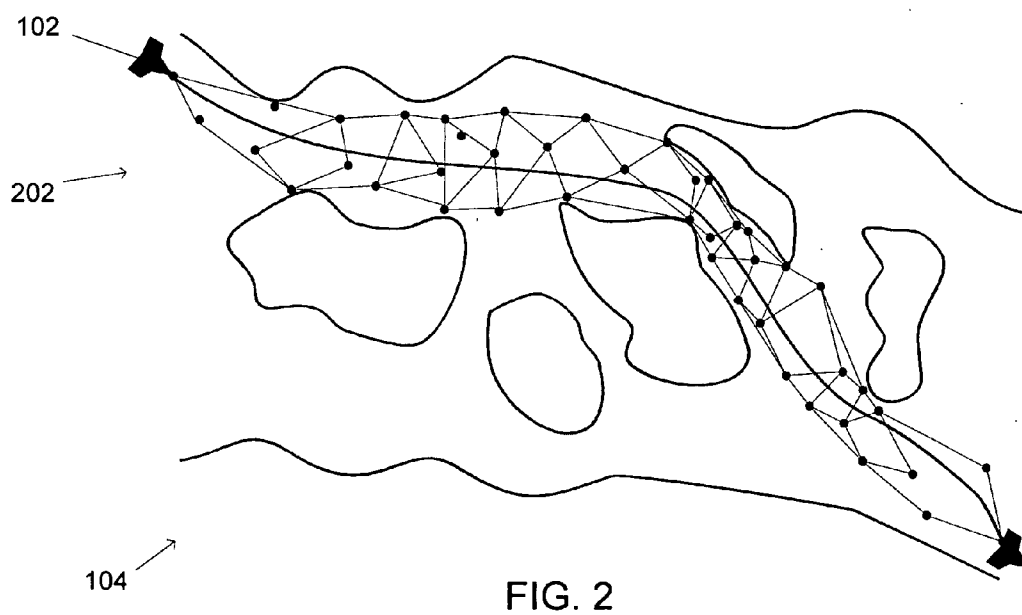
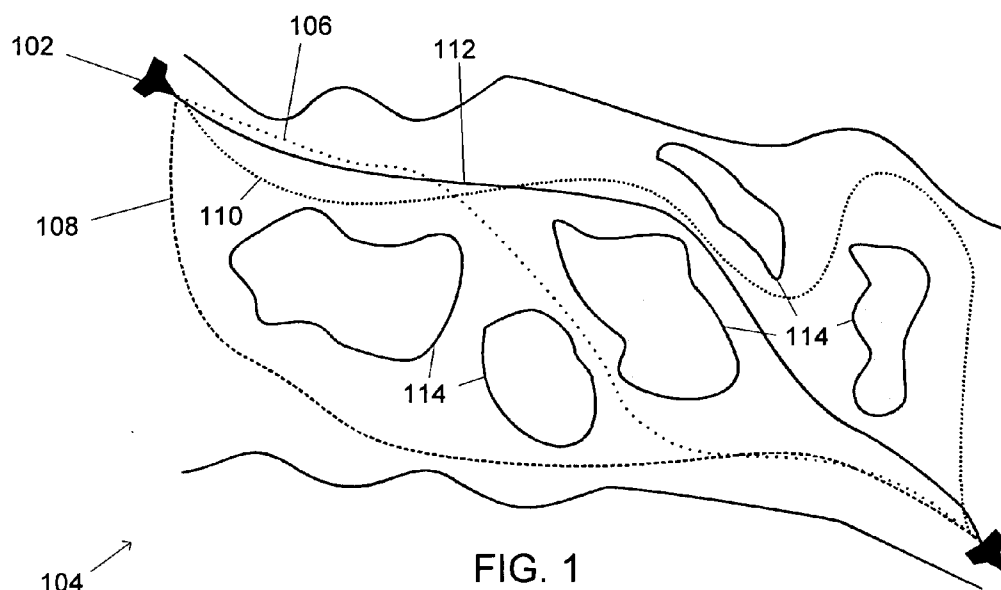
(19) **United States**(12) **Patent Application Publication**
Milam et al.(10) **Pub. No.: US 2007/0179685 A1**(43) **Pub. Date: Aug. 2, 2007**(54) **TRAJECTORY GENERATION USING
NON-UNIFORM RATIONAL B-SPLINES**(52) **U.S. Cl. 701/3; 701/23**(76) Inventors: **Mark Milam**, La Habra, CA (US);
Melvin Flores, Pasadena, CA (US)(57) **ABSTRACT**

Correspondence Address:

CARMEN B. PATTI & ASSOCIATES, LLC
ONE NORTH LASALLE STREET
44TH FLOOR
CHICAGO, IL 60602 (US)(21) Appl. No.: **11/540,471**(22) Filed: **Sep. 29, 2006****Related U.S. Application Data**(60) Provisional application No. 60/721,666, filed on Sep.
29, 2005.**Publication Classification**(51) **Int. Cl.**
G01C 23/00 (2006.01)

A trajectory is generated using non-uniform rational B-splines. A feasible corridor is approximated with at least one convex polytope. The at least one convex polytope is defined by a plurality of vertices. The feasible corridor comprises a region in a trajectory space that satisfies a plurality of trajectory constraints for a vehicle to pass through the trajectory space. The vehicle comprises a plurality of vehicle constraints. A non-uniform rational B-spline (NURBS) definition is constructed that comprises: a plurality of NURBS basis functions, a plurality of control points that comprise the plurality of vertices, and a plurality of weight points. The plurality of vehicle constraints are parameterized with the plurality of NURBS basis functions. At least one trajectory is generated for the vehicle through the trajectory space where the at least one trajectory lies within the feasible corridor to satisfy the plurality of trajectory constraints.





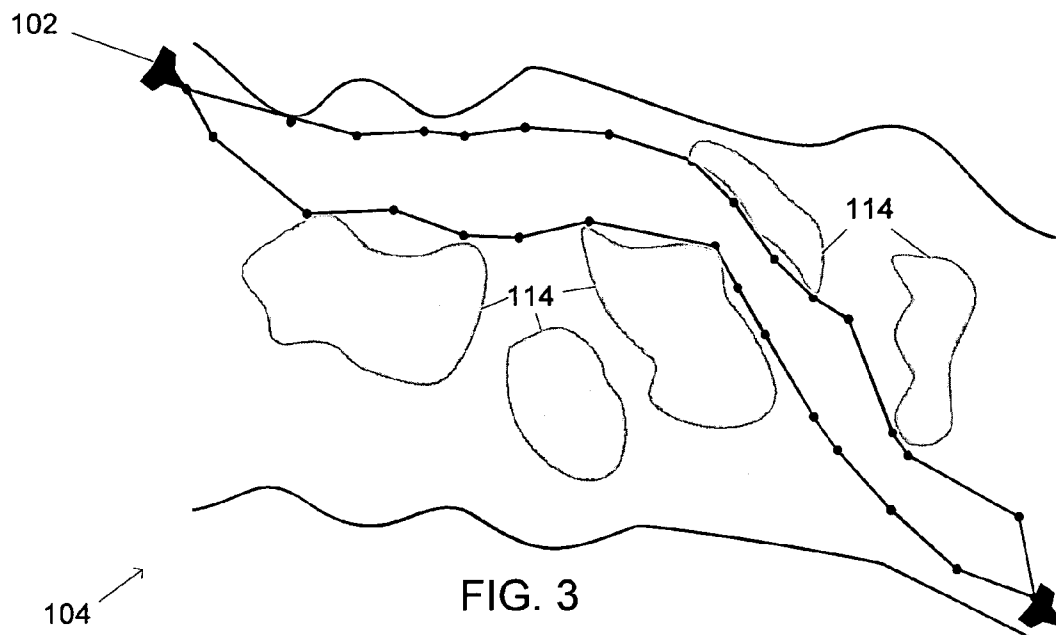


FIG. 3

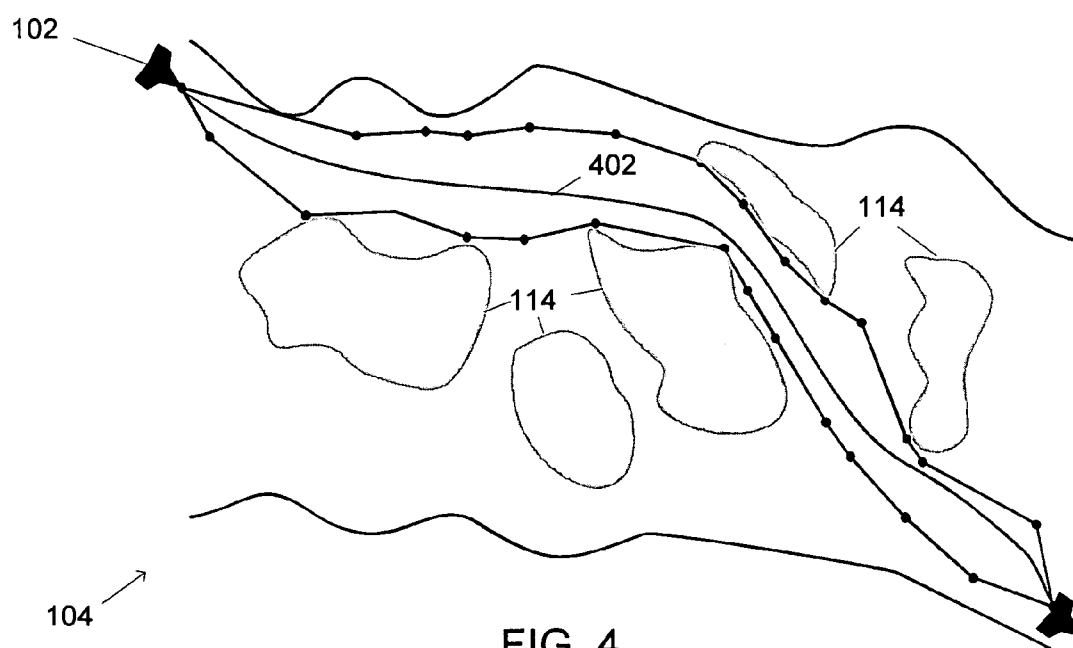
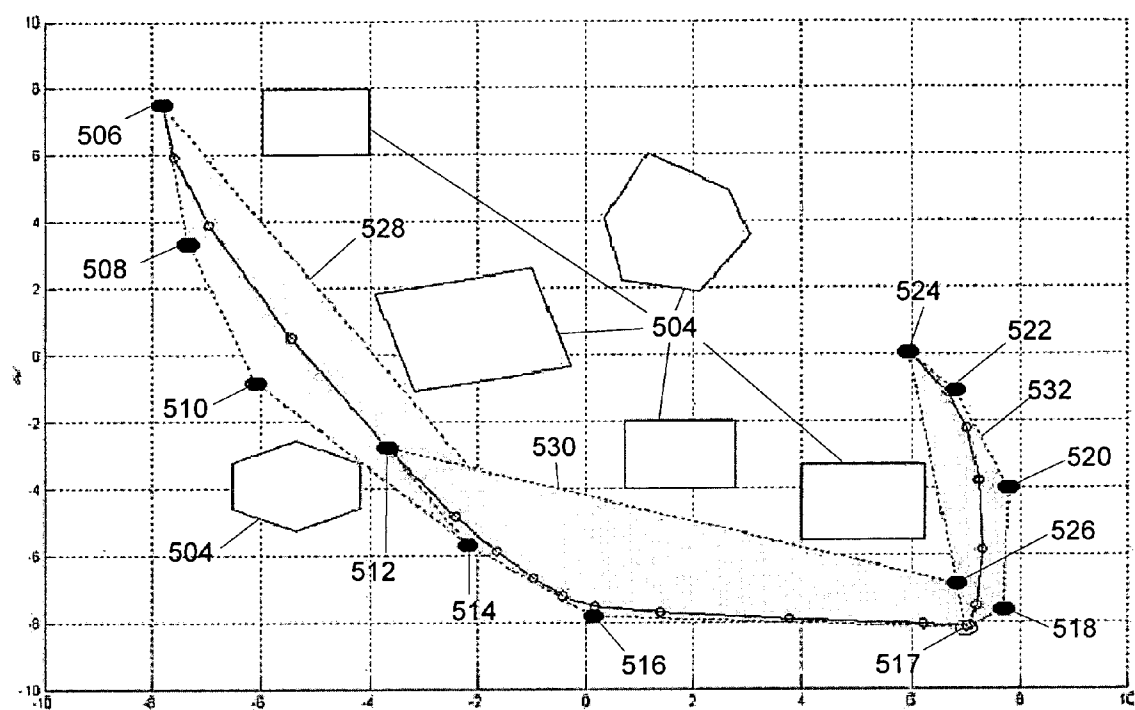


FIG. 4



502 →

FIG. 5

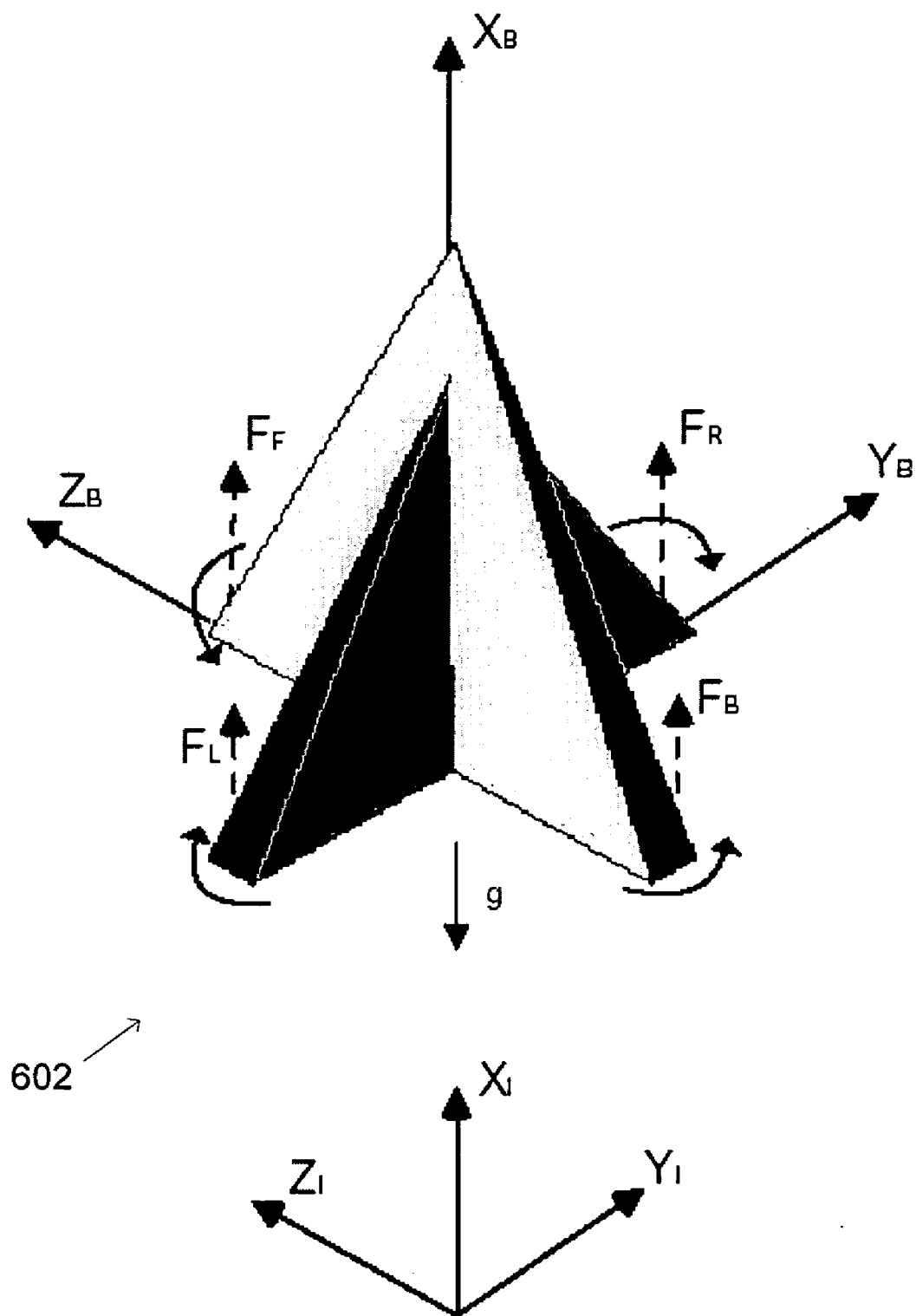


FIG. 6

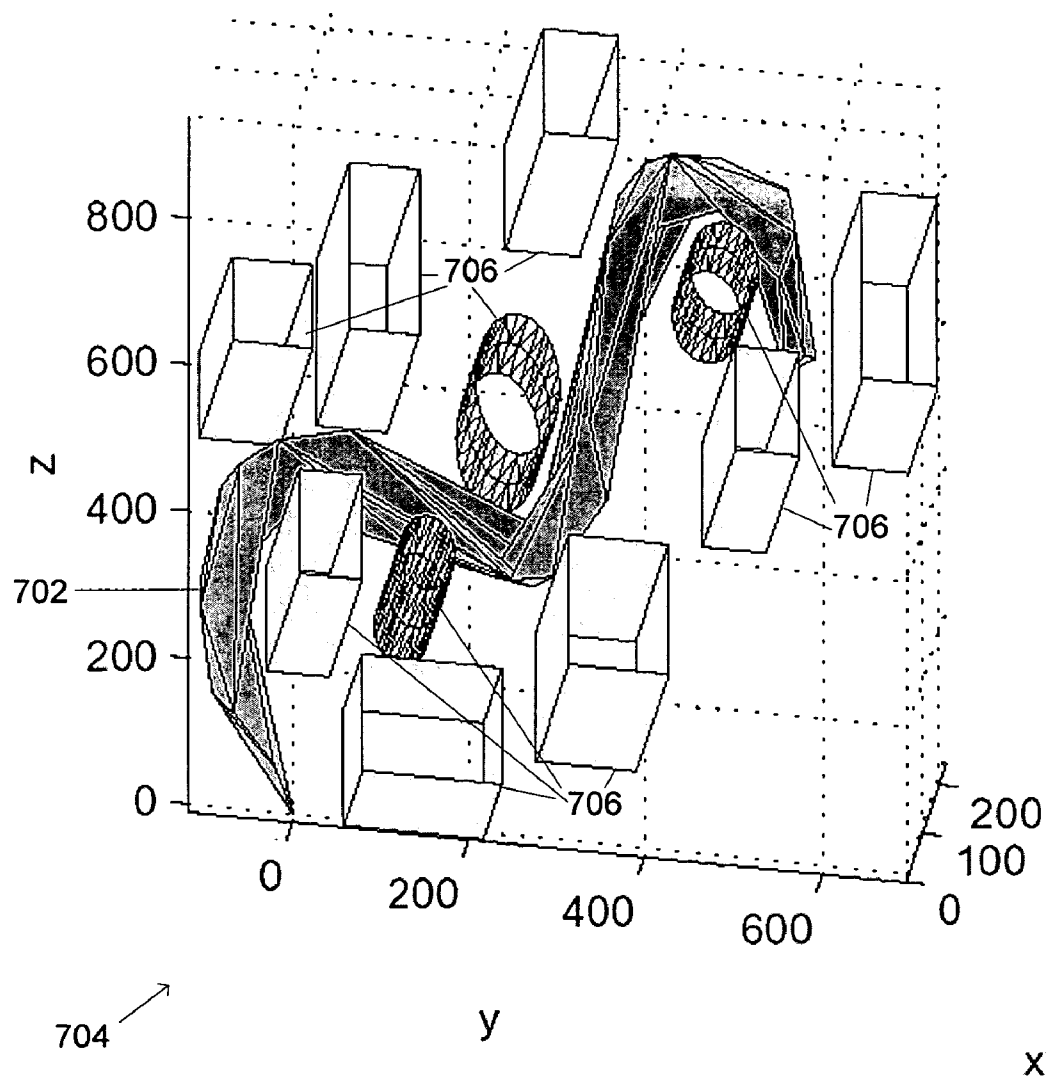


FIG. 7

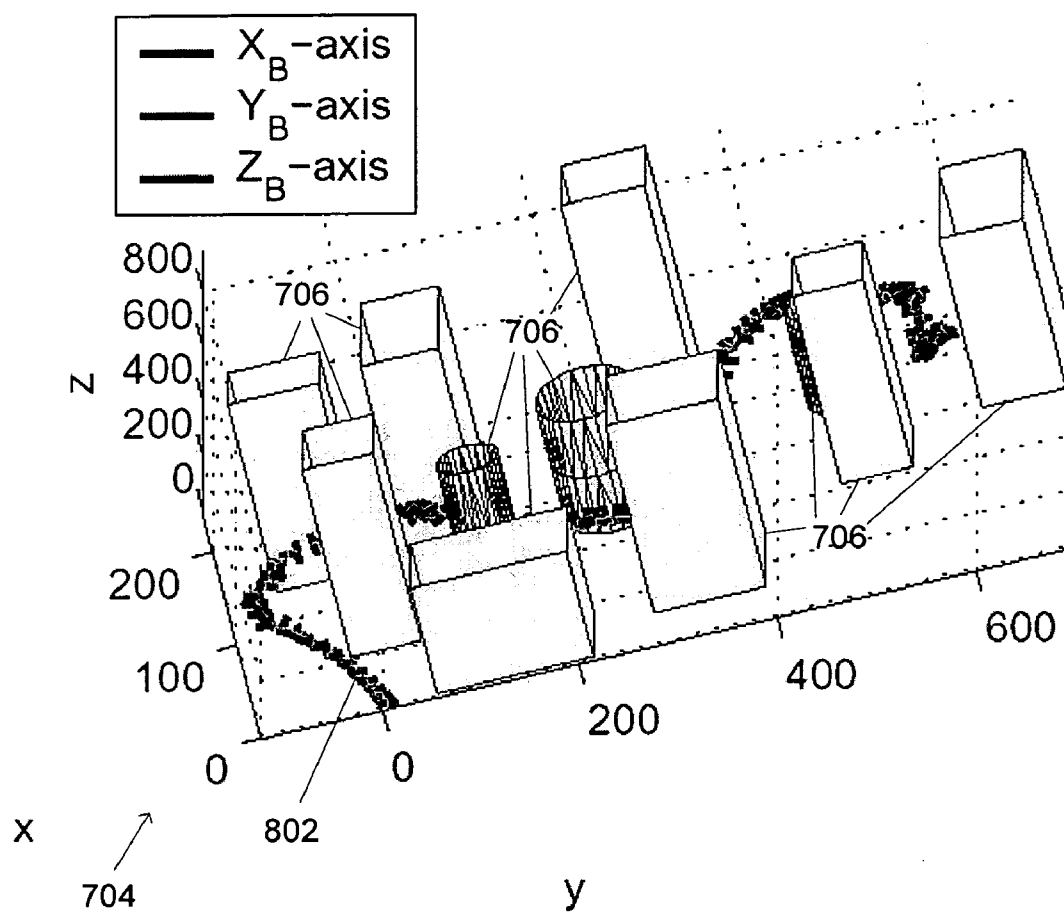


FIG. 8

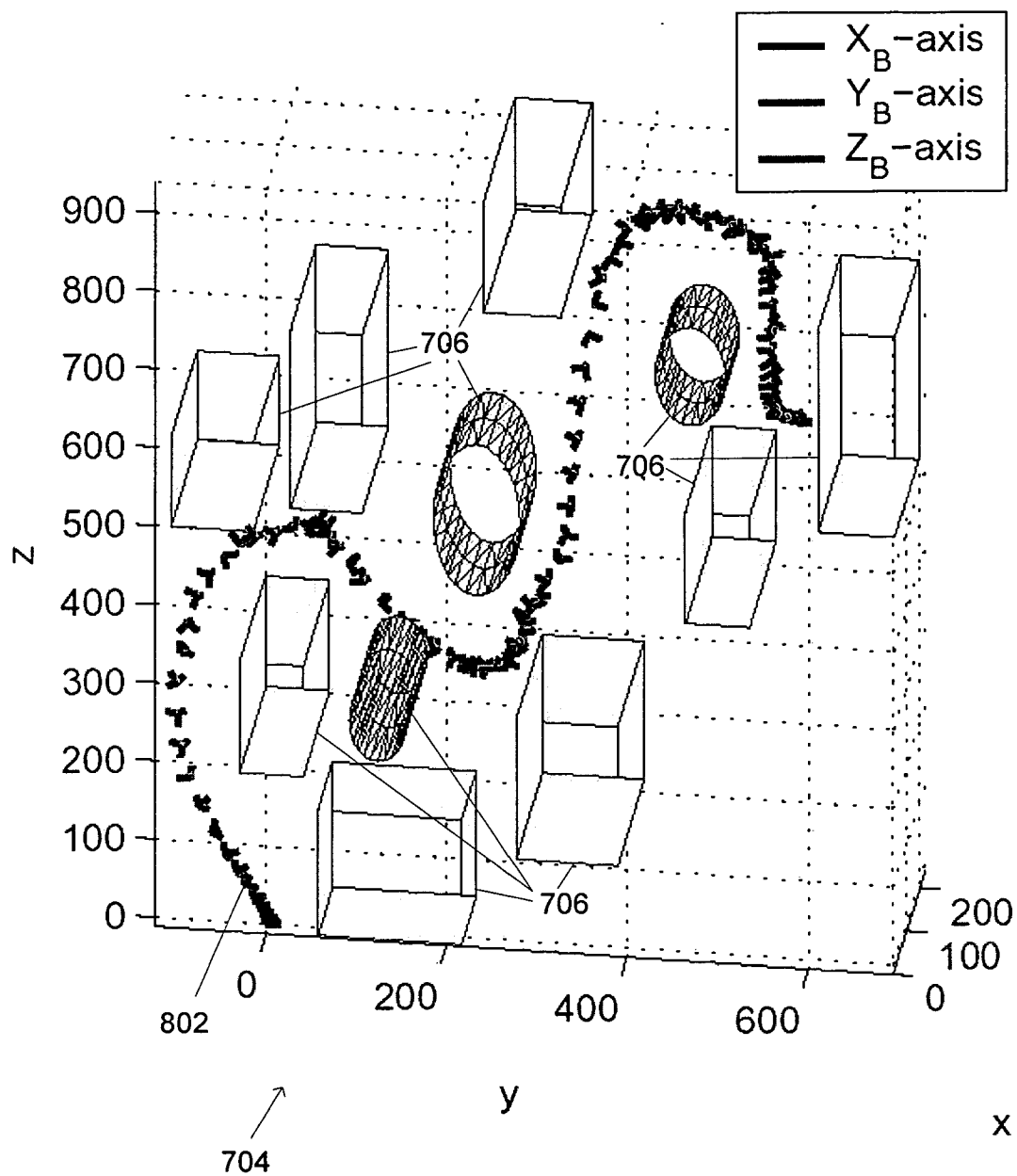


FIG. 9

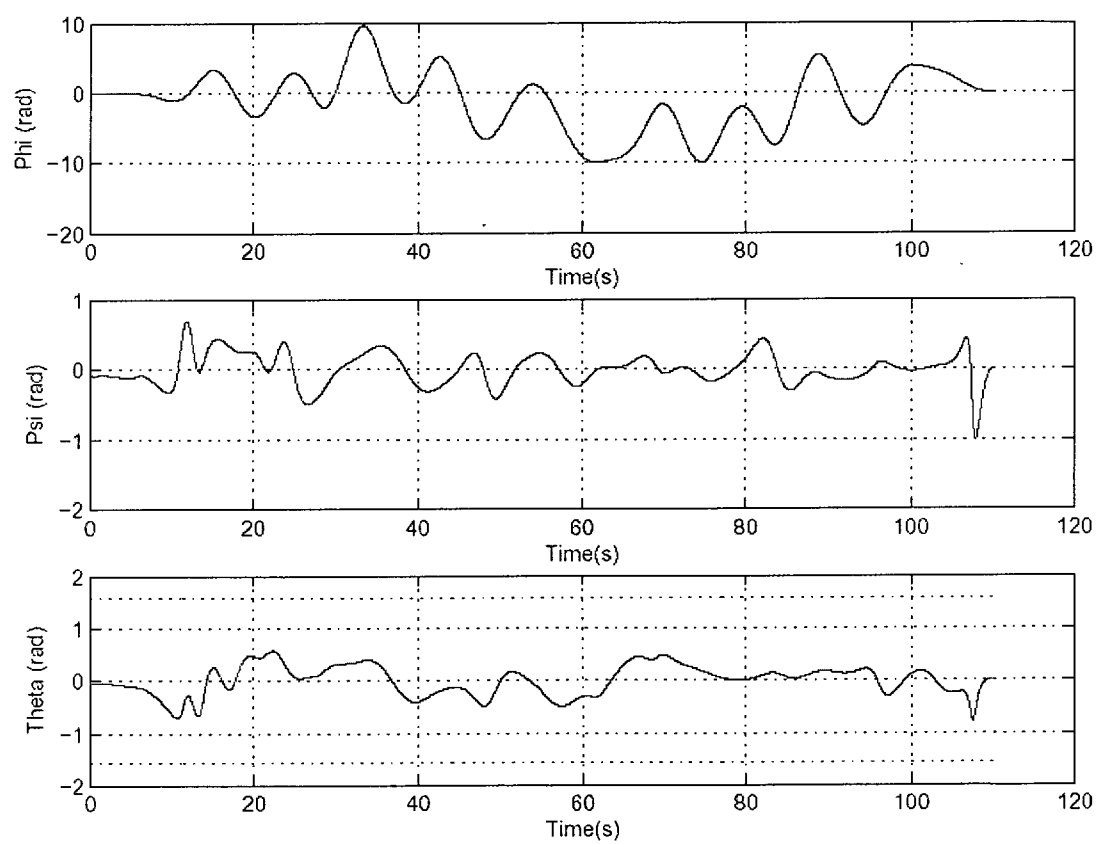


FIG. 10

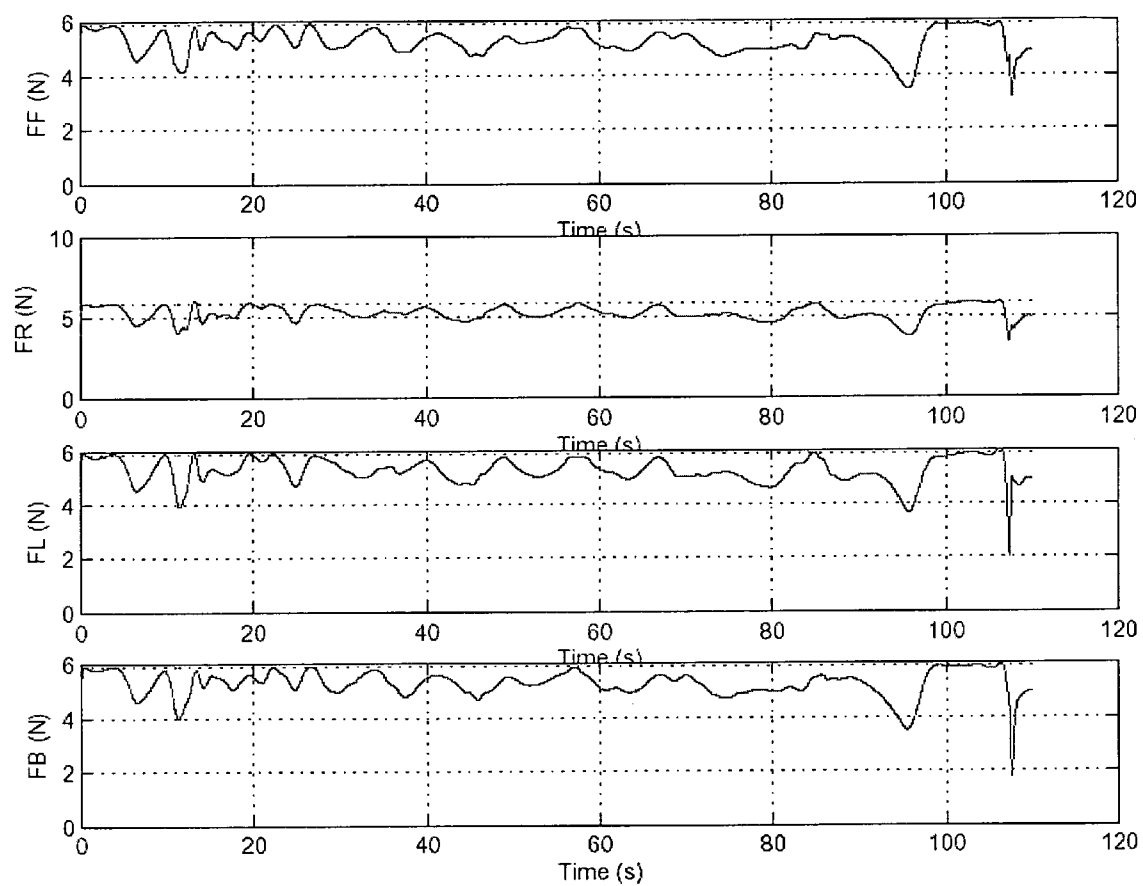


FIG. 11

TRAJECTORY GENERATION USING NON-UNIFORM RATIONAL B-SPLINES

TECHNICAL FIELD

[0001] The invention relates generally to trajectory planning and, more particularly, to trajectory planning for unmanned and manned vehicles, such as spacecraft or aerial vehicles.

BACKGROUND OF THE INVENTION

[0002] Various operations of spacecraft and other vehicles require precise trajectory planning or path planning to ensure that a vehicle proceeds to a desired destination in an optimum manner, while avoiding any obstacles. For example, an orbiting satellite may need to perform a rendezvous and proximity operation (RPO) whereby the satellite must follow a trajectory that places it in close proximity to another spacecraft for purposes of capture or maintenance of the satellite. Similarly, docking one spacecraft with another requires trajectory planning for one or both vehicles. Spacecraft re-entry and ascent guidance are other categories of trajectory planning applications. Yet another class of applications of trajectory planning is commercial aircraft routing and collision avoidance. A related class of applications is trajectory planning for autonomous unmanned aerial vehicles with terrain avoidance constraints.

[0003] In general, trajectory generation for dynamical systems consists in solving optimal control (OC) problems subject to dynamic constraints, boundary conditions, trajectory constraints, and actuator constraints. In obstacle avoidance problems, the bulk of the constraints are used to describe the obstacles, significantly increasing the amount of effort required to solve the OC problem and becoming a deterrent for the development of real-time algorithms.

[0004] For many decades now, researchers in robotics and computer science have dedicated significant effort into motion planning algorithms that concern obstacle avoidance. These efforts have concentrated in determining vehicle paths through free space at the expense of neglecting dynamic constraints or using a simplified version of them. In the area of control and dynamical systems, on the other hand, there has been significant effort to generate trajectories that take into account the dynamic and actuator constraints but typically neglect trajectory constraints or trivialize them. Thus, each of these approaches has its inherent limitations and there is still a need for a trajectory planning technique that does not suffer from the drawbacks of the prior art or that is able to leverage the results from both efforts. The present invention satisfies this need.

SUMMARY OF THE INVENTION

[0005] The invention in one implementation encompasses a method. A feasible corridor is approximated with at least one convex polytope. The at least one convex polytope is defined by a plurality of vertices. The feasible corridor comprises a region in a trajectory space that satisfies a plurality of trajectory constraints for a vehicle to pass through the trajectory space. The vehicle comprises a plurality of vehicle constraints. A non-uniform rational B-spline (NURBS) definition is constructed that comprises: a plurality of NURBS basis functions, a plurality of control points that comprise the plurality of vertices, and a plurality of

weight points. The plurality of vehicle constraints are parameterized with the plurality of NURBS basis functions. At least one trajectory is generated for the vehicle through the trajectory space where the at least one trajectory lies within the feasible corridor to satisfy the plurality of trajectory constraints.

[0006] Another implementation of the invention encompasses a method for generation of a trajectory for a vehicle through a trajectory space by solving an optimal control problem defined by at least one trajectory constraint for the trajectory space and at least one dynamic constraint for the vehicle. The optimal control problem comprises a differentially flat system. The optimal control problem is rewritten in terms of flat outputs of the differentially flat system to remove the at least one dynamic constraint. A feasible corridor is determined for the vehicle through the trajectory space. The feasible corridor is approximated through employment of a non-uniform rational B-spline (NURBS) parameterization to remove the at least one trajectory constraint. The optimal control problem is rewritten as a non-linear programming problem with weights of the NURBS parameterization as decision variables. The non-linear programming problem is solved to generate a local optimal trajectory within the feasible corridor.

[0007] Yet another implementation of the invention encompasses a method. An optimal control problem is rewritten in terms of flat outputs of the optimal control problem to obtain a modified control problem. A feasible corridor is defined with respect to at least one trajectory constraint through employment of a plurality of control points of the NURBS basis functions. The flat outputs of the modified control problem are parameterized by piecewise polynomial functions using non-uniform rational B-spline (NURBS) basis functions. The modified control problem is transcribed to a non-linear programming problem with weights of the NURBS basis functions as decision variables.

BRIEF DESCRIPTION OF THE DRAWINGS

[0008] FIG. 1 is a representation of a diagram depicting the global determination of all possible feasible corridors in trajectory space, and the selection of an optimal corridor with respect to a desired objective.

[0009] FIG. 2 is a representation of a diagram depicting a NURBS parameterization to construct an inner approximation to the optimal corridor of FIG. 1.

[0010] FIG. 3 is a representation of a diagram depicting trajectory generation optimization within the optimal corridor constructed in FIG. 2, without the need to explicitly consider trajectory constraints.

[0011] FIG. 4 is a representation of a diagram depicting the generation of a local optimal trajectory that is also feasible with respect to dynamic and actuator constraints, by using NURBS weights as decision variables.

[0012] FIG. 5 is a representation of a trajectory space for a vehicle that satisfied trajectory constraints.

[0013] FIG. 6 is a representation of coordinate frames for a vertical takeoff and landing (VTOL) vehicle.

[0014] FIG. 7 is a representation of a feasible corridor for the VTOL vehicle through a trajectory space with respect to obstacles in the trajectory space.

[0015] FIG. 8 is a representation of a front view of a trajectory for the VTOL vehicle through the trajectory space.

[0016] FIG. 9 is an aerial view of the trajectory for the VTOL vehicle through the trajectory space.

[0017] FIG. 10 is a representation of Euler angles of the trajectory for the VTOL vehicle through the trajectory space.

[0018] FIG. 11 is a representation of body forces of the trajectory for the VTOL vehicle through the trajectory space.

DETAILED DESCRIPTION OF THE INVENTION

[0019] Trajectory generation for dynamical systems consists of solving optimal control (OC) problems subject to dynamic constraints, boundary conditions, trajectory constraints and actuator constraints. In obstacle avoidance problems, the bulk of the constraints are used to describe the obstacles or rather free space, significantly increasing the amount of effort required to solve the OC problem and becoming a deterrent for the development of real-time algorithms.

[0020] For many decades now, researchers in robotics and computer science have dedicated significant effort into motion planning algorithms that concern obstacle avoidance. These efforts have concentrated in determining vehicle paths through free space at the expense of neglecting dynamic constraints or using a simplified version of them [8]. In the area of control and dynamical systems, on the other hand, there has been significant effort to generate trajectories that take into account the dynamic and actuator constraints but typically neglect trajectory constraints or trivialize them. Thus, each of these approaches has its inherent limitations and there is still a need for a trajectory planning technique that does not suffer from the drawbacks of the prior art or that is able to leverage the results from both efforts.

[0021] The present approach endeavors to bridge the gap between the above approaches and solve OC problems with full generality. The novelty of the approach resides in part in the ability to divide the trajectory generation procedure into two steps: a) construction of a feasible region with respect to trajectory constraints using convex polytopes and b) generation of local optimal trajectories mindful of such construction, which are guaranteed to remain inside the prescribed region and which are also feasible with respect to dynamic and actuator constraints. FIGS. 1-4 illustrate a procedure by which this approach could be applied to an unmanned aerial vehicle (UAV) for terrain avoidance or urban reconnaissance: a) a global solver is used to determine feasible corridors through trajectory space and to isolate an optimal corridor with respect to some objective, b) the corridor is inner approximated by the union of a set of convex polytopes whose vertices arise from a NURBS definition c) the inner approximation in b) allows for the removal of explicit trajectory constraints during trajectory generation, significantly reducing the effort required to solve the OC problem and d) a local optimal trajectory living in the feasible region with respect to trajectory constraints is generated which is also feasible with respect to dynamic and actuator constraints. The last step may fail to succeed due to various causes, among them, the infeasibility of the posed OC problem. In this event, apart from reviewing the OC

problem itself, the above procedure may be reiterated starting from b) and inner approximate one of the other alternative routes leading to the final position.

[0022] Traditional numerical methods for generating locally optimal trajectories for dynamical systems may be classified into two general approaches: direct and indirect. Indirect methods are developed through the calculus of variations and arise in the form of first-order and second-order necessary and sufficient conditions of optimality [2] [15][12][17]. Direct methods, on the other hand, rely on the direct transcription of the OC problem to a Nonlinear programming (NLP) problem via parameterization of the inputs and states, followed by a discretization of the resulting OC problem [1][17][9]. The reader is referred to [1] for a survey on numerical methods for trajectory generation, including their pros and cons. A third approach (hybrid approach) aims at developing methods that exploit the advantages of both of the above approaches [6][17].

[0023] Real-time solution of constrained OC problems is considered via a direct method. Although this approach extends to more general problems, this example relates to differentially flat systems and considers a piecewise polynomial parameterizations of the decision variables of the OC problem using NURBS basis functions. Differentially flat systems include controllable linear systems, and nonlinear systems which are feedback linearizable either by static or dynamic feedback [5]. Moreover, an aircraft in forward flight and a class of vertical take-off and landing (VTOL) aircraft as well as fully and certain underactuated spacecraft may be approximated by differentially flat systems. Trajectory generation for differentially flat systems has been conducted by various authors more notably [11], [4] and [10]. In [10], the issue of real-time trajectory generation was tackled successfully by the use of a direct method and B-spline parameterization of the flat outputs. In [4], the feasible region was approximated by a single polytope. Needless to say, depending on the number of obstacles and their position in space, this can be very conservative. In this example, both of these results are extended by retaining the real-time capabilities of [10] and allowing the inner approximation of the feasible region with respect to trajectory constraints by a union of convex polytopes. In order to relieve the burden often associated with direct methods, the properties of both differentially flat systems and NURBS basis functions are exploited to induce a transformation of the original OC problem into a simpler, more favorable numerical computational form. This is accomplished by effectively removing the dynamic and trajectory constraints from the original OC problem. As previously determined by other authors [5], for differentially flat systems there exists a set of flat outputs (equal in number to the inputs) such that all states and inputs can be determined from these outputs without integration. Consequently, the OC problem may be rewritten in terms of the flat outputs and then find a minimizer in the so-called flat-output space. Since the flat outputs implicitly contain all the information about the dynamics of the system, by introducing this transformation, no explicit dynamic constraints remain in the transformed OC problem (removal of dynamic constraints). Further parameterization of these flat outputs by NURBS basis functions allows exploitation of the fact that NURBS basis functions depend on two sets of parameters (control points and weights) and that one set of parameters (control points) may be used to specify, off-line, a region of space that

automatically satisfies the trajectory constraints and which, in addition, guarantees that the generated trajectories will never leave this region (removal of trajectory constraints).

[0024] After these transformations, the original OC problem has been converted into a modified OC problem without dynamic and trajectory constraints. The modified OC problem is then transcribed to a Nonlinear Programming (NLP) problem with the remaining parameters (weights) as the decision variables. In summary, this approach consists of the following steps: a) Re-write the original OC problem in terms of the flat outputs (removal of dynamic constraints) b) Parameterize the flat outputs by piecewise polynomial functions using NURBS basis functions. c) Use the control points to specify off-line a region of space which is feasible with respect to trajectory constraints (removal of trajectory constraints) d) Transcribe the modified OC problem to an NLP problem with the weights as the decision variables. The present invention endeavors to solve optimal control (OC) problems with full generality. This is possible by optimizing with respect to the trajectory constraints separately from optimization with respect to the dynamic and actuator constraints.

[0025] A principal object of the present invention is to generate local optimal trajectories for dynamic systems, which are guaranteed to be feasible with respect to trajectory constraints, by parameterizing decision variables by Non-Uniform Rational B-Splines (NURBS) and judiciously exploiting their properties. In particular, the invention uses this parameterization to define a feasible corridor (a union of convex polytopes constructed by a finite set of vertices) a priori, guaranteeing that trajectories generated, thereafter, will remain inside the prescribed region.

[0026] The novelty of the approach resides in part in the ability to divide the trajectory generation procedure into two steps: a) construction of a feasible region with respect to trajectory constraints using convex polytopes and b) generation of local optimal trajectories mindful of such construction, which are guaranteed to remain inside the prescribed region and which are also feasible with respect to dynamic and actuator constraints.

[0027] One embodiment of the method of the invention may be defined as comprising the steps of (a) determining all feasible corridors through trajectory space and determining the optimal corridor with respect to a desired objective; (b) using a NURBS (non-uniform rational B-splines) parameterization to construct an approximation to the optimal corridor; (c) generating trajectories that 'live' in the optimal corridor, without explicitly considering trajectory constraints; and then (d) generating a local optimal trajectory also living in the optimal corridor. The local optimal trajectory is also feasible with respect to dynamic and actuator constraints. The resulting trajectory is, therefore, guaranteed to remain inside the prescribed region of feasible corridors, and is feasible with respect to trajectory constraints as well as dynamic and actuator constraints.

[0028] As illustrated by way of example in the drawings, the present invention is concerned with trajectory planning, which may be applied in various contexts, such as space vehicle control in rendezvous, docking, launch and reentry procedures, or in the control of terrestrial or airborne vehicles. In the past, trajectory planning techniques have focused either on consideration of trajectory constraints

(usually in the form of obstacles), or on consideration of dynamic and actuator constraints associated with the vehicle or object being controlled. In accordance with the present invention, trajectory generation is divided into two distinct steps, one of which constructs a feasible region of trajectories that satisfy the trajectory constraints, and the other of which generates a local optimal trajectory that falls within the feasible region and is also feasible with respect to dynamic and actuator constraints.

[0029] The accompanying drawings in FIGS. 1-4 illustrate an embodiment of the invention as applied to an unmanned aerial vehicle (UAV) 102. Recall that one goal of the invention is to generate trajectories for dynamical systems that avoid obstacles present in trajectory space.

[0030] As indicated in FIG. 1, a global solver is used to determine all feasible corridors through a trajectory space 104 and to single out the optimal corridor with respect to some objective. The feasible corridors in this example comprise corridors 106, 108, 110, and 112. The global solver takes into account the trajectory constraints, such as obstacles 114, and possibly some simplified version of the dynamics of the vehicle. In order to obtain a global solution, the structure of the optimal control (OC) problem is exploited (typically, convexity). The various curved lines passing between adjacent obstacles illustrate several of the possible paths between a starting point and a destination point for the vehicle. The solid line represents the selected optimal path in this example.

[0031] FIG. 2 illustrates that a NURBS parameterization is used to construct an approximation to the optimal region determined in accordance with FIG. 1. The introduction of NURBS basis functions allows the use of two types of parameters: weights and control points. The control points are used for approximating the optimal region isolated by the steps shown in FIG. 1, while the weights are used later to generate trajectories that are guaranteed to never leave the prescribed region. This step of the method exploits the fact that piecewise polynomial functions defined by NURBS basis functions are guaranteed to remain inside the convex hull of the control points defining each polynomial piece. As a result of smoothness requirements imposed on the decision variables, these convex sets (convex hulls) overlap, producing a connected region 202 as shown in FIG. 2. The union of convex polytopes in FIG. 2 approximates the optimal corridor.

[0032] The feasible corridor constructed in accordance with FIG. 2 allows the generation of trajectories that 'live' in this region and it simplifies the trajectory generation optimization by not being required to explicitly consider the trajectory constraints as depicted in FIG. 3. The specification of a feasible corridor in this way allows removal of the explicit trajectory constraints from the optimization.

[0033] Finally, a local optimal trajectory 402 living in this region is generated, as indicated in FIG. 4. This local optimal trajectory 402 is also made feasible with respect to dynamic and actuator constraints, by using the NURBS weights as decision variables as shown. The optimal trajectory that also is feasible with respect to the all other remaining constraints is obtained by trajectory generation. The resulting curve is guaranteed to remain inside the prescribed region.

[0034] An illustrative description of operation of the method is presented, for explanatory purposes. Consider the

following general OC problem subject to control and state constraints:

$$\min_{x,u} J[x, u] = g_0(x(t_0), u(t_0)) + \int_{t_0}^{t_f} g_t(x(t), u(t))dt + g_f(x(t_f), u(t_f)) \quad (1)$$

Subject to:

$$\dot{x} = F(x(t), u(t)), t \in [t_0, t_f] \quad (2)$$

$$l_0 \leq A_0 x(t_0) + B_0 u(t_0) \leq u_0$$

$$l_t \leq A_t x(t) + B_t u(t) \leq u_t, t \in [t_0, t_f]$$

$$l_f \leq A_f x(t_f) + B_f u(t_f) \leq u_f \quad (3)$$

$$L_0 \leq c_0(x(t_0), u(t_0)) \leq U_0$$

$$L_t \leq c_t(x(t), u(t)) \leq U_t, t \in [t_0, t_f]$$

$$L_f \leq c_f(x(t_f), u(t_f)) \leq U_f$$

where $x(t) \in \mathbb{R}^n$ and $u(t) \in U \subset \mathbb{R}^m$. Furthermore, assume that all the involved functions are sufficiently smooth and of the appropriate dimensions.

[0035] Removal of Dynamic Constraints

[0036] As mentioned before, we will only consider differentially flat systems. In this particular case, there exist flat outputs $z \in \mathbb{R}^m$ of the form

$$z = \Psi(x, u^{(0)}, u^{(1)}, \dots, u^{(r)}) \quad (4)$$

such that states and inputs can be written as Follows:

$$z = \gamma(Z_1, \dots, Z_m) \quad (5)$$

$$u = \mu(Z_1, \dots, Z_m) \quad (6)$$

where $Z_1 = (Z_1^{(0)}, \dots, Z_1^{(D_1)}) \in \mathbb{R}^{N_1^{17}}$. Since the behavior of flat systems is determined by the flat outputs, we can plan trajectories in flat space and then map these to appropriate inputs.

[0037] With this transformation, the OC problem can be rewritten as follows:

$$\min_z J[z] = G_0(z(t_0)) + \int_{t_0}^{t_f} G_t(z(t))dt + G_f(z(t_f))$$

Subject to:

$$l_0 \leq A_0 Z(t_0) \leq u_0$$

$$l_t \leq A_t Z(t) \leq u_t, t \in [t_0, t_f]$$

$$l_f \leq A_f Z(t_f) \leq u_f$$

$$L_0 \leq C_0(Z(t_0)) \leq U_0$$

$$L_t \leq C_t(Z(t)) \leq U_t, t \in [t_0, t_f]$$

$$L_f \leq C_f(Z(t_f)) \leq U_f$$

where $X: [t_0, t_f] \rightarrow \mathbb{R}^{N^u}$ is the total number of variables required for the problem; $Z(t) = (Z_1(t), \dots, Z_m(t)) \in \mathbb{R}^{N^u}$.

[0038] Parameterization of Flat Outputs

[0039] The search for minimizers is then restricted to the space of piecewise polynomial functions, P . More specifically, the search is restricted to functions with a prescribed number of polynomial pieces, order and smoothness. This space is identified by $P_{b,o,s}$, where $b = (b_0, \dots, b_{N_p})$ are the (N_p+1) breakpoints, specifying the sites at which the end-

points of the N_p polynomial pieces of order o being joined with smoothness $s_j = s, j=1, \dots, N_p-1$ reside. This space is finite dimensional and it can be efficiently represented in terms of B-spline basis functions [3]. The dimension of the space is obtained by

$$\dim(P_{b,o,s}) = N_c = N_{p,o} - \sum_{i=0}^{N_p} (s_i + 1) \quad (7)$$

[0040] Consequently, a curve $c(t) \in P_{b,o,s}$ can be expressed in terms of the B-splines basis functions as follows:

$$c^{(r)}(t) = \sum_{i=0}^{N_c-1} B_{j,d}^{(r)}(t) p_j, \quad t \in [t_0, t_f], \quad r = 0, \dots, s \quad (8)$$

[0041] where $B_{j,d}(t)$ is the j th B-spline basis function of degree d and p_j is the corresponding j th control point. The B-spline basis functions are computed recursively:

[0042] For degree =0:

$$B_{j,0}(t) = \begin{cases} 1 & \text{if } t \in [k_j, k_{j+1}) \\ 0 & \text{otherwise} \end{cases}$$

[0043] For degree >0:

$$B_{j,d}(t) = \frac{t - k_j}{k_{j+d} - k_j} B_{j,d-1}(t) + \frac{k_{j+d+1} - t}{k_{j+d+1} - k_{j+1}} B_{j+1,d-1}(t)$$

and the r th time derivative of the basis function $B_{j,d}(t)$ is given by:

$$B_{j,d}^{(r)}(t) = d \left(\frac{B_{j,d-1}^{(r-1)}(t)}{k_{j+d} - k_j} - \frac{B_{j+1,d-1}^{(r-1)}(t)}{k_{j+d+1} - k_{j+1}} \right)$$

[0044] where k_j are the non-periodic knotpoints. That is, the knotpoints are constructed by repeating the breakpoint j , m_j times. m_j is the multiplicity of the particular breakpoint and is defined by $m_j = d - s_j$ in general. In this case, the end breakpoints are discontinuous ($m_j = d+1, j=\{0, N_p\}$) and the internal breakpoints are s continuous ($m_j = d - s, j=\{1, \dots, N_p-1\}$). The non-periodic knotpoints then take the form:

$$k = (b_0^{d+1}, \dots, b_0^{d-1}, \dots, b_1^{d+1}, \dots, b_1^{d-s}, \dots, b_{N_p}^{d+1}, \dots)$$

[0045] Instead of B-spline basis functions, NURBS basis functions are used, which are themselves defined in terms of B-splines. Therefore, all the B-spline definitions above are required for their development. A curve $c(t)$ is expressed in terms of NURBS basis functions as follows:

$$c^{(r)}(t) = \sum_{j=0}^{N_c-1} R_{j,d}^{(r)}(t) p_j, \quad t \in [t_0, t_f] \quad (9)$$

[0046] where $R_{j,d}^{(r)}(t)$ is the r th time derivative of the j th NURBS basis function of degree d and p_j are the corresponding j th control points. The NURBS basis functions are expressed in terms of B-spline basis functions:

$$R_{j,d}(t) = \frac{B_{j,d}(t)w_j}{\sum_{i=0}^{N_c-1} B_{i,d}(t)w_i} \quad (10)$$

[0047] where $w_j > 0$ is the j th weight corresponding to the j th control point. By introducing, NURBS basis functions the number of decision parameters available is increased: weights in addition to control points. This increase allows the use of control points for defining a feasible region with respect to the trajectory constraints while using the weights to generate feasible trajectories that are guaranteed to never leave this region.

[0048] NURBS basis functions have many useful properties. In particular, they are nonnegative, satisfy the partition of unity property, have local support, remain in the convex hull of the control points, and all their derivatives exist in the interior of a knot span, $[k_i, k_{i+1})$, where they are rational functions with nonzero denominator [14].

[0049] Parameterizing the flat outputs and their derivatives in this manner provides:

$$\tilde{z}_i(t) = \begin{bmatrix} R_{0,d_i}^{(0)}(t) & \dots & R_{N_c-1,d_i}^{(0)}(t) \\ \vdots & \ddots & \vdots \\ R_{0,d_i}^{(D_i)}(t) & \dots & R_{N_c-1,d_i}^{(D_i)}(t) \end{bmatrix} \begin{bmatrix} p_0^i \\ \vdots \\ p_{N_c-1}^i \end{bmatrix}$$

$\tilde{R}_{d_i}(t, w^i) \quad p^i$

and

$$\tilde{z}(t) = \begin{bmatrix} \tilde{R}_{d_1}(t, w^1) & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \tilde{R}_{d_m}(t, w^m) \end{bmatrix} \begin{bmatrix} p^1 \\ \vdots \\ p^m \end{bmatrix}$$

$\tilde{R}(t, \tilde{w}) \quad \tilde{p}$

where $\tilde{w} = (w^1, \dots, w^m)$

[0050] Removal of Trajectory Constraints

[0051] In general, trajectories that avoid obstacles present in trajectory space are desirable. Some of the constraints defined thus far in the OC problem are of this type. One goal is to construct a corridor through free space. That is, a region defined in such a way that any trajectory in it avoids such obstacles, or rather, it satisfies the trajectory constraints. Advantageously, piecewise polynomial functions defined by NURBS basis functions are guaranteed to remain inside the convex hull of the control points defining each polynomial

piece. As a result of smoothness requirements imposed on the flat outputs, these convex sets (convex hulls) overlap, producing a connected region. This method makes use of two important properties of NURBS: the convex hull property and the local support property. As a consequence, the region being defined by the control points will be a union of N_p convex hulls. The goal then is to position the control points in such a way that the generated set specifies a region through trajectory space that satisfies all the trajectory constraints (avoids all obstacles), see FIG. 5. Since the resulting trajectory, if it exists, is guaranteed to remain inside this region, these constraints can effectively be removed from the OC Problem. The weights associated with these control points will be used as decision variables to generate a trajectory inside this region.

[0052] Referring to FIG. 5, a trajectory space 502 comprises a plurality of obstacles 504. A plurality of control points 506, 508, 510, 512, 514, 516, 517, 518, 520, 522, 524, and 526 form three overlapping convex hulls 528, 530, and 532. For example, convex hull 528 comprises control points 506, 508, 510, 514, and 516, convex hull 530 comprises control points 512, 514, 516, 517, 518, and 526, and convex hull 532 comprises control points 517, 518, 520, 522, 524, and 526. In this example, a desired trajectory for the UAV 102 may begin at point 506 and end at point 524.

[0053] Discretization of Modified OC Problem

[0054] Consider the following uniform time partition:

$$T_0 = T_0 < T_1 < \dots < T_{N_T-1} < T_{N_T} = t_f$$

[0055] where

$$\tau_i = t_0 + i\Delta\tau, \quad i = 0, \dots, N_T - 1, \quad \Delta\tau = \frac{t_f - t_0}{N_T - 1}$$

[0056] Then a discrete version of the optimal control problem may be written as follows:

$$\min_z J[\tilde{z}] = G_0(\tilde{z}(\tau_0)) + \sum_{i=0}^{N_T-2} \left[\sum_{j=0}^N \gamma_j^i G_i(\tilde{z}(\tau_j)) \right] + G_f(\tilde{z}(\tau_{N_T-1}))$$

Subject to:

$$\begin{aligned} l_0 &\leq \tilde{A}_0 \tilde{Z}(\tau_0) \leq u_0 \\ l_i &\leq \tilde{A}_i \tilde{Z}(\tau_i) \leq u_i, i=0, \dots, N_T-1 \\ l_f &\leq \tilde{A}_f \tilde{Z}(\tau_{N_T-1}) \leq u_f \\ L_0 &\leq \tilde{C}_0(\tilde{Z}(\tau_0)) \leq U_0 \\ L_i &\leq \tilde{C}_i(\tilde{Z}(\tau_i)) \leq U_i, i=0, \dots, N_T-1 \\ L_f &\leq \tilde{C}_f(\tilde{Z}(\tau_{N_T-1})) \leq U_f \end{aligned}$$

[0057] In addition, the integral has been approximated by a quadrature formula. For this purpose the nodes $\tau_j = \tau_1 + j\Delta\tau_i$ for $j = 0, 1, \dots, N$, and

$$\Delta\tau_i = \frac{\tau_{i+1} - \tau_i}{N},$$

are defined, where N is the number of points required by the quadrature formula being used.

[0058] At this point, all the B-spline basis functions can be obtained at the respective points of the time partition and the OC problem has only the weights and the control points as unknowns. Recall that since some of the control points have been used to describe a feasible region through trajectory space, these control points are fixed and will not be used as decision variables. However the corresponding weights will be used to generate the trajectory traversing the specified region.

[0059] Nonlinear Programming Problem

[0060] Effectively after all these transformations, the original OC problem has been transcribed into an NLP problem. At this point, any NLP solver can be used to solve this problem. For this example, the SNOPT solver [7] has been used. SNOPT uses a Sequential Quadratic Programming (SQP) algorithm and it has been optimized to solve general Sparse NLP problems cast in the following form:

$$\min_y f(y)$$

Subject to:

$$L \leq \begin{bmatrix} y \\ C(y) \\ Ay \end{bmatrix} \leq U$$

[0061] in this case the unknowns are the weights and control points defining the flat outputs; therefore,

$$y = \begin{bmatrix} \tilde{p} \\ \tilde{w} \end{bmatrix}$$

[0062] the cost function

$$f(y) = G_0(\tilde{z}(\tau_0)) + \sum_{i=0}^{N_t-2} \left[\sum_{j=0}^N \gamma_j^i G_i(\tilde{z}(r_j)) \right] + G_f(\tilde{z}(\tau_{N_t-1}))$$

[0063] the nonlinear constraints

$$\begin{bmatrix} L_0 \\ L_t \\ \vdots \\ L_t \\ L_f \end{bmatrix} \leq \begin{bmatrix} \tilde{C}_0(\tilde{z}(\tau_0)) \\ \tilde{C}_t(\tilde{z}(\tau_0)) \\ \vdots \\ \tilde{C}_t(\tilde{z}(\tau_{N_t-1})) \\ \tilde{C}_0(\tilde{z}(\tau_{N_t-1})) \end{bmatrix} \leq \begin{bmatrix} U_0 \\ U_t \\ \vdots \\ U_t \\ U_f \end{bmatrix}$$

[0064] and the linear constraints

$$\begin{bmatrix} l_0 \\ l_t \\ \vdots \\ l_t \\ l_f \end{bmatrix} \leq \begin{bmatrix} \tilde{A}_0 \tilde{z}(\tau_0) \\ \tilde{A}_t \tilde{z}(\tau_0) \\ \vdots \\ \tilde{A}_0 \tilde{z}(\tau_{N_t-1}) \\ \tilde{A}_f \tilde{z}(\tau_{N_t-1}) \end{bmatrix} \leq \begin{bmatrix} u_0 \\ u_t \\ \vdots \\ u_t \\ u_f \end{bmatrix}$$

[0065] In order to obtain a minimizer for the NLP problem, the Jacobians of the involved functions must be calculated with respect to the unknowns. In general, for any function $F(\tilde{Z}(\tau_j))$ at any time τ_j its Jacobian is obtained by using the chain rule to obtain:

$$D_y F(\tilde{Z}(\tau_j)) = D_Z F(\tilde{Z}(\tau_j)) D_Y \tilde{Z}(\tau_j)$$

[0066] The partial derivatives with respect to each of these parameters are expressed in the form

$$\frac{\partial z_i(t, w^i, p^i)}{\partial t} = \frac{\partial}{\partial t} \left[\frac{\sum_{j=0}^{N_t-1} B_{j,d_i}(t) w_j^i p_j^i}{\sum_{i=0}^{N_t-1} B_{i,d_i}(t) w_i^i} \right]$$

$$D_{w^i} z_i(t, w^i, p^i) = \left[\frac{\partial z_i(t, w^i, p^i)}{\partial w_0^i} \quad \dots \quad \frac{\partial z_i(t, w^i, p^i)}{\partial w_{N_t-1}^i} \right]$$

$$D_{p^i} z_i(t, w^i, p^i) = \left[\frac{\partial z_i(t, w^i, p^i)}{\partial p_0^i} \quad \dots \quad \frac{\partial z_i(t, w^i, p^i)}{\partial p_{N_t-1}^i} \right]$$

[0067] where the individual derivatives are obtained from

$$z_i^{(r)}(t, w^i, p^i) = \frac{\sum_{j=0}^{N_t-1} B_{j,d_i}^{(r)}(t) w_j^i p_j^i}{\sum_{j=0}^{N_t-1} B_{j,d_i}(t) w_j^i} - \sum_{l=1}^r \binom{r}{l} \Psi_l^i z_i^{(r-l)}(t, w^i, p^i)$$

$$\frac{\partial z_i^{(r)}(t, w^i, p^i)}{\partial w_k^i} = \Phi_i^r - \Phi_i^0 z_i^{(r)}(t, w^i, p^i) -$$

$$\sum_{l=1}^r \binom{r}{l} \Phi_l^i z_i^{(r-l)}(t, w^i, p^i) - \sum_{l=1}^r \binom{r}{l} \Psi_l^i \frac{\partial z_i^{(r-l)}(t, w^i, p^i)}{\partial w_k^i}$$

$$\frac{\partial z_i^{(r)}(t, w^i, p^i)}{\partial p_k^i} = \Phi_i^r - \sum_{l=1}^r \binom{r}{l} \Psi_l^i \frac{\partial z_i^{(r-l)}(t, w^i, p^i)}{\partial p_k^i}$$

$$\text{where } \Psi_l^i = \frac{\sum_{j=0}^{N_t-1} B_{j,d_i}^{(l)}(t) w_j^i}{\sum_{j=0}^{N_t-1} B_{j,d_i}(t) w_j^i}, \quad \Phi_i^l = \frac{B_{i,d_i}^{(l)}(t)}{\sum_{j=0}^{N_t-1} B_{j,d_i}(t) w_j^i}.$$

EXAMPLE

VTOL UAV

[0068] In this section, a combination of FORTRAN and C++ is used for implementation of the method to plan trajectories for a small four-propeller vertical take-off and landing (VTOL) unmanned aerial vehicle (UAV) 602. Assuming a flat-Earth approximation and near-hover dynamics (aerodynamic forces and moments are negligible), Newton's law and Euler's law for the VTOL UAV in local-level coordinates and body coordinates may be written, respectively, as follows:

$$\begin{aligned} \left[\frac{dV_B^{F_E}}{dt} \right]^L &= \frac{1}{m} [T^T]^{BL} [F_p]^B - [g]^L \\ \left[\frac{d\omega^{F_B/F_E}}{dt} \right]^B &= [I_B^{F_B-1}]^B [M_p]^B - [I_B^{F_B-1}]^B [\Omega^{F_B/F_E}]^B [I_B^{F_B}]^B [\omega^{F_B/F_E}]^E \\ \text{where} \\ [V_B^{F_E}]^L &= \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix}, \quad [g]^L = \begin{bmatrix} g \\ 0 \\ 0 \end{bmatrix}, \quad [F_p]^B = \begin{bmatrix} F \\ 0 \\ 0 \end{bmatrix} \\ [\omega^{F_B/F_E}]^B &= \begin{bmatrix} p \\ q \\ r \end{bmatrix}, \quad [\Omega^{F_B/F_E}]^B = \begin{bmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{bmatrix} \\ [I_B^{F_E}]^B &= \begin{bmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{bmatrix}, \quad [M_p]^B = \begin{bmatrix} M_{x_B} \\ M_{y_B} \\ M_{z_B} \end{bmatrix} \\ [T^T]^{BL} &= \begin{bmatrix} c\theta c\psi & s\theta s\phi c\psi - s\psi c\phi & s\theta c\phi c\psi + s\psi s\phi \\ c\theta s\psi & s\psi s\theta s\phi + c\psi c\phi & s\psi s\theta c\phi - c\psi s\phi \\ -s\theta & s\phi c\theta & c\phi c\theta \end{bmatrix} \end{aligned}$$

[0069] with c and s being used instead of cos and sin, correspondingly. (x,y,z) denotes the position of the center of mass of the UAV. The magnitude of applied force along the X_B body axis is denoted F. The mass of the UAV is $m=2.0$ kg, g is gravity, and p, q, and r are the components of the angular velocity in body coordinates. Moreover, it is assumed that there are no products of inertia. The principle moments of inertia are denoted as follows: $I_x=0.25$ kg m², $I_y=0.125$ kg m², and $I_z=0.125$ kg m². The equations of motion become in these coordinates:

$$\begin{aligned} \begin{bmatrix} \ddot{x} + \dot{y} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} &= \frac{F}{m} \begin{bmatrix} \cos\theta \cos\psi \\ \cos\theta \sin\psi \\ -\sin\theta \end{bmatrix} \\ \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} &= \begin{bmatrix} (1/I_x)[M_{x_B} + (I_y - I_z)qr] \\ (1/I_y)[M_{y_B} + (I_z - I_x)pr] \\ (1/I_z)[M_{z_B} + (I_x - I_y)pq] \end{bmatrix} \end{aligned}$$

[0070] Consider the minimum time OC problem of taking the VTOL UAV from an initial equilibrium solution to a final equilibrium solution subject to obstacle constraints and input constraints.

$$\min_{x,u} \int_{t_0}^{t_f} dt \quad (11)$$

$$\dot{x} = f(x, u) \quad (12)$$

$$(x(t_0), x(t_f)) \in B \quad (13)$$

$$(x(t), y(t), z(t)) \in S \quad (14)$$

$$u(t) \in U \quad (15)$$

[0071] where the states $x=(x, \dot{x}, y, \dot{y}, z, \dot{z}, \Phi, \dot{\Phi}, \theta, \dot{\theta}, \Psi, \dot{\Psi})$ and the inputs $u=(F_L, F_R, F_F, F_B)$. This system is differentially flat. The flat outputs are:

$$z_1=x, z_2=y, z_3=z, z_4=\Phi$$

[0072] The flat outputs may be written in terms of the states, inputs, and their derivatives as follows:

$$\begin{aligned} z_1^{(1)} &= \dot{x}, z_1^{(2)} = F(\cos\theta \cos\Psi) - g \\ z_2^{(1)} &= \dot{y}, z_2^{(2)} = F(\cos\theta \sin\Psi) \\ z_3^{(1)} &= \dot{z}, z_3^{(2)} = -F \sin\theta \\ z_4^{(1)} &= \dot{\Phi}, z_4^{(2)} = f_1(\Psi, \Phi, \theta, \dot{\Psi}, \dot{\Phi}, M_{x_B}, M_{y_B}, M_{z_B}) \\ z_1^{(3)} &= f_2(F, \dot{F}, \Psi, \Phi, \theta, \dot{\Psi}, \dot{\Phi}) \\ z_2^{(3)} &= f_3(F, \dot{F}, \Psi, \Phi, \theta, \dot{\Psi}, \dot{\Phi}) \\ z_3^{(3)} &= f_4(F, \dot{F}, \Psi, \Phi, \theta, \dot{\Psi}, \dot{\Phi}) \\ z_1^{(4)} &= f_5(F, \ddot{F}, \ddot{\Psi}, \ddot{\Phi}, \Psi, \Phi, \theta, \dot{\Psi}, \dot{\Phi}, M_{x_B}, M_{z_B}) \\ z_2^{(4)} &= f_6(F, \ddot{F}, \ddot{\Psi}, \ddot{\Phi}, \Psi, \Phi, \theta, \dot{\Psi}, \dot{\Phi}, M_{y_B}, M_{z_B}) \\ z_3^{(4)} &= f_7(F, \ddot{F}, \ddot{\Psi}, \ddot{\Phi}, \Psi, \Phi, \theta, \dot{\Psi}, \dot{\Phi}, M_{y_B}, M_{z_B}). \end{aligned}$$

[0073] In short,

$$\begin{aligned} (z_1^{(1)}, \dots, z_2^{(4)}, z_2, \dots, z_2^{(4)}, z_3, \dots, z_3^{(4)}, z_4, \dots, \\ z_4^{(2)}) = \Psi(\xi), \end{aligned}$$

[0074] where

$$\xi = (x, y, z, \theta, \Phi, \Psi, \dot{z}, \dot{y}, \dot{z}, \dot{\theta}, \dot{\Phi}, \dot{\Psi}, M_{x_B}, M_{y_B}, M_{z_B}, F, \dot{F}, \ddot{F}).$$

[0075] The above relation is locally invertible, with the exception of a few points, since

$$\det \left(\frac{\partial \Psi}{\partial \xi} \right) = \frac{-F^6 \cos\theta}{I_x I_y I_z} \quad (16)$$

[0076] is nonzero. For implementation purposes, the states and inputs must be explicitly written in terms of the flat outputs. Given that

$$x=z_1, y=z_2, z=z_3, \Phi=z_4$$

[0077] and consequently

$$\dot{x} = \dot{z}_1, \quad \dot{y} = \dot{z}_2, \quad \dot{z} = \dot{z}_3, \quad \dot{\Phi} = \dot{z}_4$$

$$x^{(r)} = z_1^{(r)}, \quad y^{(r)} = z_2^{(r)}, \quad z^{(r)} = z_3^{(r)}, \quad \Phi^{(r)} = z_4^{(r)}$$

[0078] Using the equations of motion, $\Psi, \theta, F, M_{B_x}, M_{B_y}$ and M_{B_z} must then be written in terms of the flat outputs. Using Newton's law in flat-output space, Ψ, θ , and F can be solved analytically

$$\psi = \tan^{-1}\left(\frac{z_2}{z_1 + g}\right) \quad (17)$$

$$\theta = \tan^{-1}\left(\frac{-z_3}{(z_1 + g)\cos\psi + z_2\sin\psi}\right) \quad (18)$$

$$F = m\sqrt{(z_1 + g)^2 + z_2^2 + z_3^2} \quad (19)$$

[0079] Likewise, using Euler's law in flat-output space, the propulsive moments M_{Bx} , M_{By} , and M_{Bz} can be solved for as:

$$M_{Bx} = I_x \ddot{p} + (I_z - I_y) \dot{q} \dot{r} \quad (20)$$

$$M_{By} = I_y \dot{q} + (I_x - I_z) \dot{p} \dot{r} \quad (21)$$

$$M_{Bz} = I_z \dot{r} + (I_y - I_x) \dot{p} \dot{q} \quad (22)$$

[0080] To write the moments in flat-output space, the components of the angular velocity in terms of Euler angles are written as:

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} \dot{\phi} - \dot{\psi} \sin\theta \\ \dot{\theta} \cos\phi + \dot{\psi} \cos\theta \sin\phi \\ \dot{\psi} \cos\theta \cos\phi - \dot{\theta} \sin\phi \end{bmatrix} \quad (23)$$

[0081] and after taking their time derivatives and substituting in equations (20)-(22):

$$M_{Bx} = f_{Bx}(I_x, I_y, I_z, \theta, \dot{\phi}, \dot{\psi}, \ddot{\phi}, \ddot{\psi})$$

$$M_{By} = f_{By}(I_x, I_y, I_z, \theta, \dot{\phi}, \dot{\psi}, \ddot{\phi}, \ddot{\psi})$$

$$M_{Bz} = f_{Bz}(I_x, I_y, I_z, \theta, \dot{\phi}, \dot{\psi}, \ddot{\phi}, \ddot{\psi})$$

[0082] The following variables are used: $\Phi, \dot{\Phi}, \ddot{\Phi}, \Psi, \dot{\Psi}, \ddot{\Psi}, \theta, \dot{\theta}, \ddot{\theta}$. Since Φ is flat output,

$$\Phi = Z_4, \dot{\Phi} = \dot{Z}_4, \ddot{\Phi} = \ddot{Z}_4$$

[0083] The other derivatives can be obtained by taking time derivatives of equations (17) and (18). After these substitutions, the moments will depend only on the flat outputs and their time derivatives. Moreover, the individual fan forces F_L , F_R , and F_B may be obtained by inverting

$$\begin{bmatrix} M_{xB} \\ M_{yB} \\ M_{zB} \\ F \end{bmatrix} = \begin{bmatrix} k_w & k_w & -k_w & -k_w \\ 0 & 0 & l_{FB} & -l_{FB} \\ l_{LR} & -l_{LR} & 0 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} F_L \\ F_R \\ F_F \\ F_B \end{bmatrix}$$

[0084] The applied thrust forces are fixed in the body as shown in FIG. 6. The torque constant of the propeller is $k_w = 0.5$, $l_{FB} = 1.0$ m is half the distance between the applied force F_F and F_B , and $l_{LR} = 1.0$ m is half the distance between the applied force F_L and F_R as seen in FIG. 6.

[0085] After this transformation, equation (12) may be removed from the minimum OC problem. The flat outputs may then be parameterized by NURBS basis functions. In this example, piecewise polynomial functions $P_{b,o,s}$ are chosen for each flat output. More specifically Z_1, \dots, Z_3 are

$P_{b,7,4}$, with 16 polynomial pieces being pasted at the break-points b , where b is constructed by evenly positioning the end points of the polynomials in the interval $[0, 1]$. The flat output Z_4 is also constructed with 16 polynomials and the same breakpoints b except for a different order and curve smoothness: $P_{b,6,3}$.

[0086] The next step is to construct a corridor **702** in S (trajectory space **704**) which avoids all obstacles **706**. This allows removal of equation (14) from the OC problem. FIG. 7 shows the corridor **702** that has been designed for this example.

[0087] The only constraints remaining in this OC problem are equations (13) and (15). The body forces F_F , F_R , F_L and F_B are constrained to be in the interval $[0, 0.3 \text{ m g}]$. The angle θ was further constrained to the interval $[-\pi/2, \pi/2]$ to avoid the singularity described in equation (16). The optimal trajectory **802** was found starting from a random initial guess as shown in FIG. 8 (front view) and FIG. 9 (aerial view). The **109.97** second trajectory was solved in real time on a 3 GHz Pentium IV PC. The euler angles and body forces are shown in FIG. 10 and FIG. 11, respectively.

[0088] Conclusion

[0089] In conclusion, a new local trajectory generation method is introduced that exploits the intrinsic properties of both differentially flat systems and NURBS Basis functions to define OC problems that are amenable to real-time solution. The ability to decouple the trajectory generation problem from feasibility with respect to space obstacles allows the design of feasible regions a priori, guaranteeing that trajectories generated thereafter will remain inside the prescribed region and, in addition, are feasible (or infeasible) with respect to dynamic and actuator constraints. Moreover, the method is successfully applied to generate trajectories for a VTOL UAV that avoids obstacles in space. The results may be extended to take advantage of a global solver, to single out global corridors that are optimal with respect to a desired objective and then define regions in such corridors to generate feasible trajectories with respect to obstacle constraints. In addition, there exists the possibility of using such a method to design receding horizon controllers for systems with state constraints.

[0090] References

[0091] [1] J. T. Betts, Survey of Numerical Methods for Trajectory Optimization, J. of Guidance, Control, and Dynamics, 21(2):193-207, 1998.

[0092] [2] A. E. Bryson, Applied Optimal Control, Washington: Hemisphere Publishing Company, 1975.

[0093] [3] C. De Boor, A Practical Guide to Splines, New York: Springer-Verlag, 1978.

[0094] [4] N. Faiz, S. K. Agrawal and R. M. Murray, Trajectory Planning of Differentially Flat Systems with Dynamics and Inequalities, J. of Guidance, Control and Dynamics, 24(2):219-227, 2001.

[0095] [5] M. Fliess, J. Levine, Ph. Martin and P. Rouchon, Flatness and defect of nonlinear systems: introductory theory and examples, Int. J. Control, 61(6):1327-1361, 1995.

[0096] [6] P. F. Gath, "CAMTOS-A software suite combining Direct and Indirect Trajectory Optimization Methods". Universitat Stuttgart Dissertation 2002.

[0097] [7] P. Gill and W. Murray and M. Saunders, "User's Guide for SNOPT 7: A Fortran Package for Large-Scale Nonlinear Programming". Systems Optimization Laboratory, Stanford University, Stanford, Calif. 94305.

[0098] [8] J. C. Latombe, Robot Motion Planning, Boston: Kluwer Academic Publishers, 1991.

[0099] [9] D. Kraft, On Converting Optimal Control Problems into Nonlinear Programming Codes, in: NATO ASI Series, Vol. F15, Computational Mathematical Programming, ed. K. Schittkowski, Springer, 1985.

[0100] [10] M. B. Milam, "Real-Time Optimal Trajectory Generation for Constrained Dynamical Systems". California Institute of Technology Dissertation 2003.

[0101] [11] M. J. van Nieuwstadt, "Trajectory Generation for Nonlinear Control Systems". California Institute of Technology Dissertation 1996.

[0102] [12] H. Pesch, Real-time Computation of Feedback Controls for Constrained Optimal Control Problems. Part 1: Neighboring extremals, Optimal Control Applications and Methods, 10:129-145, 1989.

[0103] [13] H. Pesch, Real-time Computation of Feedback Controls for Constrained Optimal Control Problems. Part 2: A Correction Method based on Neighboring extremals, Optimal Control Applications and Methods, 10:147-171, 1989.

[0104] [14] L. Piegle and W. Tiller, The NURBS Book, 2nd edition, Germany: Springer-Verlag, 1997.

[0105] [15] L. S. Pontryagin, V. G. Boltyanskii, R. V. Gamkrelidze and E. F. Mischchenko, The Mathematical Theory of Optimal Processes, Wiley-Interscience, 1962.

[0106] [16] M. Rathinam, "Differentially Flat Nonlinear Control Systems". California Institute of Technology Dissertation 1997.

[0107] [17] O. von Stryk and R. Bulirsch, Direct and Indirect Methods for Trajectory Optimization, Annals of Operations Research, 37:357-373, 1992.

What is claimed is:

1. A method, comprising the steps of:

approximating a feasible corridor with at least one convex polytope, wherein the at least one convex polytope is defined by a plurality of vertices, wherein the feasible corridor comprises a region in a trajectory space that satisfies a plurality of trajectory constraints for a vehicle to pass through the trajectory space, wherein the vehicle comprises a plurality of vehicle constraints;

constructing a non-uniform rational B-spline (NURBS) definition that comprises:

a plurality of NURBS basis functions,

a plurality of control points that comprise the plurality of vertices, and

a plurality of weight points;

parameterizing the plurality of vehicle constraints with the plurality of NURBS basis functions;

generating at least one trajectory for the vehicle through the trajectory space, wherein the at least one trajectory lies within the feasible corridor to satisfy the plurality of trajectory constraints.

2. The method of claim 1, further comprising the step of:

determining at least one available corridor through the trajectory space through employment of a global solver, wherein the at least one available corridor comprises the feasible corridor;

selecting the feasible corridor from the at least one available corridor.

3. The method of claim 1, wherein the plurality of vehicle constraints comprise at least one dynamic constraint that is defined by at least one decision variable;

wherein the step of parameterizing the plurality of vehicle constraints with the plurality of NURBS basis functions comprises the step of:

parameterizing the at least one decision variable in terms of the plurality of NURBS basis functions.

4. The method of claim 3, wherein the plurality of vehicle constraints comprise a differentially flat system, the method further comprising the step of:

rewriting the plurality of vehicle constraints in terms of at least one flat output of the differentially flat system.

5. The method of claim 1, wherein the step of generating the at least one trajectory for the vehicle through the trajectory space comprises the step of:

generating a local optimal trajectory within the feasible corridor.

6. The method of claim 5, wherein the feasible corridor comprises a first feasible corridor, the method further comprising the steps of:

choosing a second feasible corridor upon a failure of the generation of the local optimal trajectory;

inner approximating the second feasible corridor;

generating a local optimal trajectory within the second feasible corridor with respect to the at least one trajectory constraint.

7. The method of claim 1, wherein the step of approximating the feasible corridor with the at least one convex polytope comprises the step of:

inner approximating the feasible corridor by a union of a plurality of convex polytopes to remove the plurality of trajectory constraints, wherein the plurality of convex polytopes comprise the at least one convex polytope.

8. The method of claim 1, wherein the step of approximating the feasible corridor with the at least one convex polytope comprises the step of:

approximating the feasible corridor with a plurality of convex polytopes.

9. A method for generation of a trajectory for a vehicle through a trajectory space by solving an optimal control problem defined by at least one trajectory constraint for the trajectory space and at least one dynamic constraint for the vehicle, wherein the optimal control problem comprises a differentially flat system, the method comprising the steps of:

rewriting the optimal control problem in terms of flat outputs of the differentially flat system to remove the at least one dynamic constraint;

determining a feasible corridor for the vehicle through the trajectory space;

approximating the feasible corridor through employment of a non-uniform rational B-spline (NURBS) parameterization to remove the at least one trajectory constraint;

rewriting the optimal control problem as a non-linear programming problem with weights of the NURBS parameterization as decision variables;

solving the non-linear programming problem to generate a local optimal trajectory within the feasible corridor.

10. The method of claim 9, wherein the step of determining the feasible corridor for the vehicle through the trajectory space comprises the step of:

determining the feasible corridor for the vehicle through the trajectory space such that the at least one trajectory constraint is satisfied.

11. The method of claim 10, wherein the step of determining the feasible corridor for the vehicle through the trajectory space such that the at least one trajectory constraint is satisfied comprises the step of:

employing a global solver to determine the feasible corridor.

12. The method of claim 9, wherein the step of approximating the feasible corridor through employment of the non-uniform rational B-spline (NURBS) parameterization to remove the at least one trajectory constraint comprises the step of:

approximating the feasible corridor through employment of a plurality of piecewise polynomial functions.

13. The method of claim 12, wherein the step of approximating the feasible corridor through employment of the plurality of piecewise polynomial functions comprises the step of:

selecting at least one piecewise polynomial function such that the at least one piecewise polynomial function has a predetermined order and smoothness.

14. The method of claim 12, wherein the NURBS parameterization is defined by a plurality of control points and a plurality of weight points, wherein the step of approximating the feasible corridor through employment of the NURBS parameterization to remove the at least one trajectory constraint comprises the step of:

defining the feasible corridor by the plurality of control points.

15. The method of claim 12, wherein the NURBS parameterization is defined by a plurality of control points and a plurality of weight points, wherein the step of approximating the feasible corridor through employment of the NURBS parameterization to remove the at least one trajectory constraint comprises the step of:

defining the feasible corridor by a subset of the plurality of control points.

16. The method of claim 12, wherein the non-linear programming problem comprises a plurality of functions, wherein the step of solving the non-linear programming problem to generate the local optimal trajectory within the feasible corridor comprises the step of:

generating Jacobians for the plurality of functions.

17. A method, comprising the steps of:

rewriting an optimal control problem in terms of flat outputs of the optimal control problem to obtain a modified control problem;

defining a feasible corridor with respect to at least one trajectory constraint through employment of a plurality of control points of the NURBS basis functions;

parameterizing the flat outputs of the modified control problem by piecewise polynomial functions using non-uniform rational B-spline (NURBS) basis functions;

transcribing the modified control problem to a non-linear programming problem with weights of the NURBS basis functions as decision variables.

18. The method of claim 17, wherein the optimal control problem comprises a determination of a vehicle trajectory through a trajectory space, wherein the feasible corridor comprises a corridor through the trajectory space that satisfies the at least one trajectory constraint, the method further comprising the step of:

inner approximating the feasible corridor through the trajectory space by a plurality of convex polytopes defined by the plurality of control points.

19. The method of claim 17, wherein the optimal control problem comprises at least one dynamic constraint for the vehicle trajectory, wherein the step of rewriting the optimal control problem in terms of the flat outputs of the optimal control problem to obtain the modified control problem comprises the step of:

rewriting the optimal control problem in terms of the flat outputs of the optimal control problem to remove the at least one dynamic constraint for the vehicle trajectory.

20. The method of claim 17, wherein the step of defining the feasible corridor with respect to the at least one trajectory constraint through employment of the plurality of control points of the NURBS basis functions comprises the step of:

determining at least one available corridor through the trajectory space through employment of a global solver, wherein the at least one available corridor comprises the feasible corridor;

selecting the feasible corridor from the at least one available corridor.

* * * * *