The humanoid bipedal locomotion requires computationally efficient solutions of the navigation and inverse kinematics problems. This paper presents analytic methods, using tools from computational geometry and techniques from the theory of Lie groups, to develop new geometric algorithms for the navigation path planning, locomotion movement, and kinematics modeling of humanoid robots. To solve the global navigation problem, we introduce the new fast marching method modified (FM3) algorithm, based on the fast marching methods (FMM) used to study interface motion, that gives a close-form solution for the humanoid collision-free whole body trajectory (WBT) calculation. For the bipedal locomotion, we build the new geometric algorithm one step to goal (OSG), to produce a general solution for the body and footstep planning which make the humanoid to move a single step towards a defined objective. We develop the new approach called sagittal kinematics division (SKD), for the humanoid modeling analysis, to allow us to solve the humanoid inverse kinematics problem using the mathematical techniques of Lie groups, like the product of exponentials (POE). The works are presented along with computed examples of the humanoid robot RH0 at the University Carlos III of Madrid. We remark that this paper introduces only closed-form solutions, numerically stable and geometrically meaningful, suitable for real-time applications.


I. INTRODUCTION

The humanoid navigation and kinematics problems present a formidable computational challenge, especially for real-time applications, due to the high number of degrees of freedom (DOF), complex model, and balance constraints [9][12]. As the complexity increases, the need for more elegant formulations of the equations of motion becomes a paramount issue, and because we believe that abstraction saves time at the end, this paper presents a slightly more difficult mathematics to handle the humanoid problems, that facilitates the development of geometric solutions.

The first objective of this paper is to solve the global navigation problem for a humanoid, despite the complexity of the environment. We develop the new fast marching method modified (FM3) algorithm, to achieve the collision-free humanoid whole body trajectory (WBT) amongst any kind of obstacles. The algorithm is based on techniques introduced by Sethian [1] for evolving interfaces in computational geometry.

The second goal of this work is to develop a general solution for the bipedal locomotion [8]. We build the new one step to goal (OSG) geometric algorithm, to get the humanoid body and footstep trajectories which make the robot to move one step towards a predefined goal, taking into account the balance constraints of the zero moment point (ZMP). The method has five phases called: Orientate, tilt, elevate, lean and balance. A bipedal locomotion will be achieved by successively executing the algorithm for a series of goals.

The third aim of this paper is to solve the humanoid inverse kinematics. To facilitate the analysis, we introduce a new approach for the humanoid model called sagittal kinematics division (SKD), which considers the saggital division of the humanoid in two manipulators, these are, the left and right robots. Afterwards, we apply the theory of Lie groups as presented by Murray [2], to solve the complete inverse kinematics problem. The product of exponentials (POE) tool is the cornerstone of our development.

Then, a complex humanoid bipedal locomotion might be developed as follows: First, the FM3 gives a collision-free global trajectory, second, we use this information to solve the bipedal locomotion with the OSG, which gives the body and footstep paths, and finally, we solve for each point of those paths the inverse kinematics problem, following the humanoid SKD model, using the POE and Lie techniques.

The development of general-purpose motion generation tools has been based on the use of virtual humanoid robot platforms [7]. In a similar way, our results are tested and illustrated with a graphical environment, using the virtual reality modeling language (VRML), for the humanoid robot RH0 of the University Carlos III of Madrid, Fig.1.
II. BACKGROUND

A. Fast Marching Methods (FMM) for Robot Navigation

The FMM are numerical techniques for analyzing interaction motion. They rely on an Eulerian perspective for the view of moving boundaries. Consider a boundary separating a region from another, moving with a speed function \( F \), one way to characterize the position of this expanding front is to compute the arrival time function \( T \), as the front crosses each point of the space. Sethian [1] introduced the FMM boundary value formulation to solve the Eikonal equation (1) that characterizes the front motion.

\[
\nabla T \bigg|_F = 1
\]

A practical approach to solve the Eikonal equation comes from the viscosity solutions for the hyperbolic conservation laws. This is the difference first order space convex scheme (2), for a number \( n \) of space dimensions \( \theta \).

\[
\left[ \sum_{i=1}^{n} \left( \max(D^{-\theta}T,0)^2 + \min(D^{+\theta}T,0)^2 \right) \right]^{1 \over 2} = \frac{1}{F} \tag{2}
\]

Where the spatial derivatives \( D \) have the formulation given by (3), where \( G \) is the spatial discretization interval.

\[
\begin{align*}
D^{+\theta}T & \equiv \frac{T(\theta+G,t)-T(\theta,t)}{G} \\
D^{-\theta}T & \equiv \frac{T(\theta,t)-T(\theta-G,t)}{G}
\end{align*} \tag{3}
\]

The FMM can be applied to problems in path planning, extracting among all possible solutions, the one that corresponds to the first arrival of information from the initial disturbance, finding out the shortest possible path from the initial to the final state.

B. Lie Groups and Algebras in Robotics

The Lie groups are very important for the mathematical analysis and the geometry, because they serve to describe the symmetry of analytical structures [11]. A Lie group is an analytical manifold that is also a group. A Lie algebra is a vectorial space over a field, that captures completely the structure of the corresponding Lie group.

The homogeneous representation of a rigid motion belongs to the special Euclidean Lie group \( \text{SE}(3) \) [2]. The Lie algebra of \( \text{SE}(3) \), denoted \( \text{so}(3) \), can be identified with the matrices called twists \( \omega \times \theta \) (4), where the skew-symmetric matrix \( \omega \times \theta \) (5) is the Lie algebra \( \text{so}(3) \) of the orthogonal special Lie group \( \text{SO}(3) \), which represents all rotations in the three-dimensional space [3]. A twist can be geometrically interpreted using the screw theory [6], as Charles’s theorem proved that any rigid body motion could be produced by a translation along a line followed by a rotation around the same line, this is a screw motion, and the infinitesimal version of a screw motion is a twist.

\[
\omega = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix}, \quad \theta = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix}
\]

\[
\text{so}(3) = \{ \omega \times \theta : \omega \in \mathbb{R}^3, \theta \in \mathbb{R}^3 \} \subseteq \mathbb{R}^{3 \times 3}
\]

(4)

The main connection between \( \text{SE}(3) \) and \( \text{so}(3) \) is the exponential transformation (6). It is possible to generalize the forward kinematics map for an arbitrary open-chain manipulator with \( n \) DOF of magnitude \( \theta \), through the product of those exponentials, expressed as \( \text{POE} \) (7), where \( g(0) \) is the reference position for the coordinate system.

\[
\begin{align*}
\omega^\theta & = \begin{bmatrix} e^{\omega \phi} \\ 0 \end{bmatrix}, \quad (I - e^{\omega \phi}) (\omega \times \theta) + \omega \times \theta = \omega^\phi \\
\omega & = \begin{bmatrix} 1 & \nu \theta \\ 0 & 1 \end{bmatrix} e^{\omega \phi} \in \text{SE}(3); \omega \in \mathbb{R}^3, \theta \in \mathbb{R}^3
\end{align*} \tag{6}
\]

\[
g(\theta) = \prod_{i=1}^{n} e^{\omega^\phi \cdot g(0)} \tag{7}
\]

A very important payoff for the \( \text{POE} \) formalism is that it provides an elegant formulation of a set of canonical problems, the Paden [4] and Kahan [5] sub-problems, among others, which have a geometric solution for theirs inverse kinematics. It is possible to obtain a close-form solution for the inverse kinematics problem of complex mechanical systems by reducing them into the appropriate canonical sub-problems.

III. NAVIGATION PATH PLANNING WITH THE NEW GEOMETRIC ALGORITHM FM3

We develop the new FM3 algorithm, for generating collision-free trajectories for the humanoid WBT. Some FMM numeric methods are not computationally very efficient because of the quadratic approach (2). The FM3 overcomes this problem with the construction of a linear scheme for approximating the Eikonal equation (1), through the formulation given by (8).

\[
\max_{i=1}^{n} \left( D^{-\theta}T, D^{+\theta}T \right) = \frac{1}{F} \tag{8}
\]

This new scheme is directly resolvable for a number \( n \) of space dimensions, where the spatial derivatives \( D \) of the function \( T \) in each space dimension \( \theta \), are expressed by (9), with a spatial discretization interval equal to \( G \).

\[
D^{\pm\theta}T \equiv \frac{T(\theta \pm G, t) - T(\theta,t)}{G}
\]

(9)

The FM3 arrival function \( T \) value increases from one given origin to all points inside the configuration space, in the sense of information propagation. Graphically Fig. 3, \( T \) is the development of a plane interface in motion (kind of accumulative pyramidal fronts.) The linear FM3 approach generates a non-differentiable but continuous \( T \) function, and therefore the explicit construction for the WBT can be obtained through back propagation following the maximum negative value of the arrival function \( T \), from the final goal to the starting point of the humanoid present position.
The **FM3** method always allows us to find out a quasi-optimal humanoid WBT path, whatever the nature of the environment, the support surface and the obstacles. For instance, for the environment with a non-convex obstacle showed in Fig. 2, we can compare the performance of a typical potential field method for solving the path-planning problem, with the **FM3** performance showed in Fig. 3.

Analyzing the results for the potential field method, you can notice that the path-planning problem is not solved, because the non-convex nature of the obstacle creates such a deep local minimum for the potential function (see Fig. 2), that even using exit random paths algorithms, it is impossible to build a trajectory to reach the goal.

Conversely, the theoretical nature of the **FM3** with its always-growing function $T$ impedes the existence of local minimum, whatever the nature of the obstacles. In this example, the trajectory is straightforwardly built, and it rounds the obstacle by its left to reach the goal, as it is showed in Fig. 3.

In the humanoid global navigation problem we need to consider the availability of environment maps, the humanoid positions and the nature of the support surface and obstacles. Then, using the **FM3** it is possible to calculate the collision-free humanoid WBT feasible path. To illustrate this, in the Fig. 4, we present a scenario for the humanoid RH0 with several obstacles and the resulting quasi-optimal global navigation trajectory $Qr$ that connects the starting position with the final goal.

### IV. HUMANOID BIPEDAL LOCOMOTION WITH THE NEW GEOMETRIC ALGORITHM OSG

The idea behind the new humanoid OSG algorithm is to develop a general-purpose solution for the humanoid bipedal locomotion [10]. We design geometric solutions for the footstep and body trajectories to move the humanoid only one step towards a given goal, which is normally part of a global navigation trajectory.

The OSG method takes into account the ZMP balance constraints, the humanoid mechanical constraints, and the defined step characteristics: Step height $S_h$, step width $S_w$ and step length $S_l$. The algorithm is developed through five phases: Orientate, tilt, elevate, lean and balance. The results are the trajectory $Q_{FF}$ (10) for the flying foot configurations $q_{FFi}$, and the trajectory $Q_{CM}$ (11) for the humanoid body center of mass (CM) configurations $q_{CMj}$. The five main points reached for each trajectory can be interpolated to obtain higher granularity trajectories.

\[
Q_{FF} = \{q_{FF1}, q_{FF2}, \ldots, q_{FF3}, q_{FF4}, q_{FF5}\} \tag{10}
\]

\[
Q_{CM} = \{q_{CM1}, q_{CM2}, \ldots, q_{CM3}, q_{CM4}, q_{CM5}\} \tag{11}
\]

The OSG method begins with any present humanoid pose defined by: the CM configuration $q_{CM0}$, the robot longitudinal axis $\upsilon_l$, the ZMP position, the direction corridor $C_d$, the flying foot configuration $q_{FF0}$, the flying foot axis $\upsilon_{FF}$, the support area of the flying foot $A_{SF}$, the support area of the support foot $A_{SF}$, the local objective $q_{lo}$ and the objective direction $\upsilon_d$.

Orientate is the first OSG phase, Fig. 5. The humanoid is standing with its feet on the support surface and orientates its body towards $\upsilon_d$. The trivial task is to geometrically calculate the new configurations $q_{CM1}$ and $q_{FF1}$.
Tilt is the second OSG phase, Fig. 6. The humanoid continuous with the feet on the support surface and tilts its whole body for the ZMP to reach the AsSF. The task is to geometrically define the new configurations $q_{CM3}$ and $q_{FF2}$.

Elevate is the third OSG phase, Fig. 7. The humanoid elevates the flying foot to reach the Sh and keeps moving the ZMP along the support pattern of AsSF. The geometric task is to calculate the new configurations $q_{CM4}$ and $q_{FF3}$.

Lean is the fourth OSG phase, Fig. 8. The humanoid moves the flying foot to lean it on the support surface. At the same time, the ZMP moves along the support pattern of AsSF. It is easy to see that geometrically we can calculate the new configurations $q_{CM4}$ and $q_{FF4}$.

Balance is the fifth OSG phase, Fig. 9. The humanoid stands on its feet and balances the body in order to put the ZMP in the middle of the convex hull. Geometrically it is easy to calculate the new configurations $q_{CM5}$ and $q_{FF5}$.

V. NEW HUMANOID KINEMATICS MODEL SKD

To facilitate the inverse kinematics problem solution, we introduce the new SKD humanoid model. This approach considers the humanoid analytical division by its sagittal plane into two different mechanisms; these are the right and left manipulators. Obviously, the new model imposes to both manipulators the kinematics constraint of having the same position and orientation for their common parts: pelvis, thoracic and cervical. Therefore, the humanoid inverse kinematics problem can be solved in a much easier way, by solving the inverse kinematics problem for two separated but synchronized manipulators.

The idea behind the SKD is a divide and conquers approach that tries to resemble the brain hemispherical control over the right and left human body sagittal halves, for a human bipedal locomotion.

In the Fig. 10, we apply the SKD kinematics model to the RH0. Then, the humanoid can be analyzed as constituted by two manipulators. The right manipulator has six virtual DOF (i.e., 0v1 to 0v6) corresponding to the right foot position and orientation, and eleven real DOF (i.e., 01 to 06 and 013 to 017) corresponding to the mechanical joints. The left manipulator has six virtual DOF (i.e., 0v11 to 0v16) corresponding to the left foot position and orientation, and eleven real DOF (i.e., 012 to 017 and 018 to 021) corresponding to the mechanical joints. $S_h$ is the frame attached to the base system (i.e., the inertial or spatial reference). $S_s$ is the frame attached to the humanoid CM (i.e., the tool reference), and it is very important for the locomotion as it defines the body configuration $q_{CM0}$ that can be expressed as a transformation matrix $g_{d}(\theta)$.

The humanoid direct kinematics problem for any point of the robot body $g(\theta)$, can be expressed by the POE (12) for the right manipulator and/or by the POE (13) for the left manipulator, with only knowing the transformation for that point at the robot reference position $g(0)$, and the humanoid SKD model.

$$g(\theta) = e^{\xi_{\theta_{11}} \theta_{11}} \ldots e^{\xi_{\theta_{16}} \theta_{16}} \cdot e^{\xi_{\theta_{1}} \theta_{1}} \ldots e^{\xi_{\theta_{6}} \theta_{6}} \cdot g(0) \quad (12)$$

$$g(\theta) = e^{\xi_{\theta_{11}} \theta_{11}} \ldots e^{\xi_{\theta_{16}} \theta_{16}} \cdot e^{\xi_{\theta_{12}} \theta_{12}} \ldots e^{\xi_{\theta_{17}} \theta_{17}} \cdot g(0) \quad (13)$$
VI. HUMANOID INVERSE KINEMATICS USING LIE GROUPS TECHNIQUES

To achieve a complex humanoid bipedal locomotion, we solve the global navigation problem with the FM3 algorithm, then we apply the points of the global path as successive goals for the OSG algorithm, and the results are the configurations (i.e., $q_{CM}$, $q_{RF}$ and $q_{LF}$) trajectories (i.e., $Q_{CM}$, $Q_{RF}$ and $Q_{LF}$) for the body, right foot and left foot respectively. The feet trajectories are obtained adding the flying foot configurations in successive OSG executions. To produce the humanoid motion we have to solve the inverse kinematics problem of the right and left humanoid manipulators given by the SKD model, for each point of the humanoid body configuration $q_{CM}$ that we will express from now on as a transformation matrix $g_{st}(\theta)$.

All that remains is to find out a general solution for the inverse kinematics of both manipulators, given any $g_{st}(\theta)$. The two humanoid manipulators have almost identical kinematics, and therefore, we only need to present the general solution for one of them, and apply the same to the other. Consequently, we introduce the inverse kinematics problem solution for the RH0 right manipulator showed in the Fig. 11, through seven phases, using the theory of Lie groups and the POE formalism.

The first phase formulates the direct kinematics problem, given by the formula (14). Where the transformation at the reference position $g_{st}(\theta)$ is given by (15), the DOF axes by (16) and the twists $\xi$ by (17).

$$g_{st}(\theta) = e^{e_{st1} \theta_{st1}} \cdots e^{e_{st6} \theta_{st6}} \cdot g_{st}(0) \quad (14)$$

$$g_{st}(0) = 
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 
\end{bmatrix}$$

$$v_{st} = \begin{bmatrix}
1 \\
0 \\
1 \\
0 
\end{bmatrix} \quad \omega_{st} = \begin{bmatrix}
0 \\
1 \\
1 \\
0 
\end{bmatrix}$$

$$\xi = \begin{bmatrix}
\alpha_1 \\
\alpha_2 \\
\alpha_3 \\
\alpha_4 \\
\alpha_5 \\
\alpha_6 
\end{bmatrix} \quad \delta = \begin{bmatrix}
\alpha_1 \\
\alpha_2 \\
\alpha_3 \\
\alpha_4 \\
\alpha_5 \\
\alpha_6 
\end{bmatrix}$$

The second phase gives the virtual DOF (i.e., 0vr1 to 0vr6) solutions, from the previous execution of the OSG algorithm, because each body configuration $g_{st}(\theta)$ has an associated right foot configuration $q_{RF}$ given by (18).

$$g_{st}(\theta) = q_{CM} \in O_{CM} \iff \theta_{st} = [\psi_{st1}, \psi_{st2}, \psi_{st3}, \psi_{st4}, \psi_{st5}, \psi_{st6}] \in O_R \quad (18)$$

The third phase gives the 03 solution. We pass all known terms to the left side of the equation (14), apply both sides to the point $p$, subtract the point $k$, and apply the norm. We operate in such a way because the resulting equation (19) is only affected by 03, and therefore we can rewrite the equation as (20), that is exactly the formulation of the Paden-Kahan canonical problem for the rotation to a given distance. Therefore, we can geometrically obtain the two possible values for the variable 03.

$$e^{-e_{st1} \theta_{st1}} \cdots e^{-e_{st6} \theta_{st6}} \cdot g_{st}(0) = p \cdot k \quad (19)$$

The fourth phase gives the 01 and 02 solutions. We pass all possible terms to the left side of the equation (14) and apply both sides to the point $p$. On so doing, the resulting equation (21) is only affected by 01, 02 and 03, and therefore we can rewrite the equation as (22), that is exactly the formulation of the Paden-Kahan canonical problem for two successive rotations. Therefore, we can geometrically obtain the two possible values, for the couple of variables 01 and 02.

$$e^{-e_{st1} \theta_{st1}} \cdots e^{-e_{st6} \theta_{st6}} \cdot g_{st}(0) \cdot p = e^{k_{st1}} \cdots e^{k_{st6}} \cdot p \quad \text{Paden-Kahan-TWO} \quad (21)$$

The fifth phase gives the 04 and 05 solutions. We pass all known terms to the left side of the equation (14) and apply both sides to the point $m$. As a result of these operations, the transformed equation (23) is only affected by 04 and 05, and we can rewrite the equation like (24), that is again the formulation of the Paden-Kahan canonical problem for two successive rotations around crossing axes.

$$e^{k_{st1}} \cdots e^{k_{st6}} \cdot m = e^{k_{st4}} \cdots e^{k_{st6}} \cdot m \quad \text{Paden-Kahan-TWO} \quad (23)$$

Fig. 11 Humanoid RH0 Right Manipulator under SKD Model.
The sixth phase gives the 06 solution. We pass all known terms to the left side of the equation (14) and apply both sides to the point S. As a result, the equation is transformed into (25), that is obviously only affected by 06, and we can rewrite it as (26), that is the formulation of the Paden-Kahan canonical problem for a rotation around an axis. Therefore, we can geometrically obtain the single possible value for the variable 06.

\[
e^{\begin{pmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \\ \theta_5 \\ \theta_6 \end{pmatrix}} = e^{\begin{pmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \\ \theta_5 \\ \theta_6 \end{pmatrix}} a = S \cdot e^{\begin{pmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \\ \theta_5 \\ \theta_6 \end{pmatrix}} a (25)
\]

Through these seven phases, the right manipulator inverse kinematics problem is solved in a geometric way, and even more, we have got not only one solution but the set of all possible solutions. For instance, the right leg has eight theoretical solutions, which are captured with the approach showed in this paper (28), if they exist.

\[
Solutions = \theta_1, Double, \theta_2, Double, \theta_3, Double, \theta_4, Double, \theta_5, Single = 8 (28)
\]

The seventh phase gives the 013 to 017 solutions. The manipulator shoulder and wrist do not intervene for the locomotion and therefore 013 and 017 are zero for the analyzed movement. The other arm DOF may or may not contribute to the locomotion, helping the balance control to keep the CM as close as possible to its initial geometric position, but for achieving this behavior, we must solve the arm inverse dynamics problem, that is beyond the scope of this paper. A very simple but effective practical arm kinematics solution takes advantage of the necessary body sagittal coordination (see SKD model in Fig. 10), and the right arm DOF are made equal or proportional to their complementary left leg DOF. Therefore the values for the variables 013 to 017 are defined as (27).

\[
\theta_{13} = 0; \ \theta_{14} = \theta_6; \ \theta_{15} = \theta_6; \ \theta_{16} = \theta_{10}; \ \theta_{17} = 0 (27)
\]

After repeating exactly the same technique for the left manipulator, the complete RH0 humanoid inverse kinematics problem is totally solved, indeed.

VII. CONCLUSIONS

The navigation, locomotion and inverse kinematics problems for humanoids present a formidable computational challenge, especially for real-time applications. This paper presents analytic and geometric algorithms for solving those problems. To solve the global navigation, we introduce the new FM3 algorithm, based on computational geometry, with a close-form solution for the humanoid collision-free WBT. For the bipedal locomotion, we build the new geometric algorithm OSG, to solve the body and footstep planning. The new SKD kinematics model eases to solve the inverse kinematics problem using the mathematical techniques of Lie groups like the POE. The works are presented along with computed examples and simulations of the humanoid robot RH0 at the University Carlos III of Madrid. Some results for the experimental performance of the algorithms are showed in the table I.

![Table I](image)

The prototype algorithms used for getting the results of the table I were programmed with the MATLAB interpreted language. The execution times can be improved by the migration of the routines to a compiled language.

Nevertheless, the important facts here are, first of all that the OSG algorithm has a computational efficiency of \(\Theta(n)\), where \(n\) is number of DOF, and second of all that the FM3 algorithm has an order of convergence \(\Omega(n^a)\), where \(G\) is the spatial interval for a number of \(n\) of space dimensions.

We highlight the fact that the algorithms presented in this paper have closed-form solutions with a clear geometric meaning, and therefore we believe they can be useful to develop real-time applications.

ACKNOWLEDGMENT

We are very grateful to the professors Sethian and Murray whose research inspired the contents of this paper.

We give thanks to the University Carlos III of Madrid Robotics Lab team for the help and understanding.

We sincerely acknowledge to the CICYT-DPI, for the financial support to the humanoid RH0 project.

REFERENCES