Navigation Functions in Conformal Geometric Algebra

Geoff Fink, Carlos López Franco and Nancy Arana Daniel
CUCEI - University of Guadalajara
Email: gefink@gmail.com, \{carlos.lopez, nancy.arana\}@cucei.udg.mx

Abstract—Conformal Geometric Algebra (CGA) can greatly improve controllers by simplifying the necessary equations and by its ability to apply geometric operations to more complicated geometric entities. In this paper we extend a singularity free CGA-based angular and linear velocity controller with navigation functions. The first navigation function ensures that the object being tracked is always within the camera’s field of view. The second navigation function is the ability of the controller to avoid collisions with other objects. These navigation functions can be easily added to the CGA-based controller, experimentally ensured the desired goals and proven stable by Lyapunov.

I. INTRODUCTION

In robots with six degrees of freedom (DOF), such as air vehicles or satellites, it is important that the visual servo tracking controllers do not contain any singularities in their rotation parameterization and that the object being tracked remains in the camera’s field of view. Other controllers have tried to overcome this problem with different path-planning techniques [1], limiting the rotation angles to ±π [2] or the use of unit or even dual quaternions [3]. In a previous paper [4] we proposed a controller based on Conformal Geometric Algebra (CGA) and proved it to be stable via CGA-based Lyapunov functions. CGA provides a comprehensive mathematical framework to represent the rotation angles without singularities. Other advantages of CGA are the ability to easily represent both Euclidean and projective geometries, geometric objects such as lines, planes and spheres and to apply reflections, rotations or translations among many other operations. To further take advantage of CGA in this paper we extend the controller to include navigation functions.

A homography can be made by using many different techniques [5]–[8], but in this paper we use geometric visual servoing using spheres. Once the homography is computed, an error system based on a CGA rotor, translator and motor can be developed. The rotor is constructed directly from the rotation matrix using a robust algorithm [9] that does not contain any singularities. Next, we extended the controller with different navigation functions. The first navigation function ensures that the object being tracked remains within the camera’s field of view. The second navigation function allows the camera to avoid collisions with obstacles. All of the navigation functions are implemented using CGA.

This paper is organized as follows. In Section II, a geometric model is developed to show the different Euclidean and projective relations used to calculate the homography. In Section III, a brief description of CGA and the other relevant definitions are defined. In Section IV, an adaptive CGA-based controller is proposed and is extended with CGA-based navigation functions. In Section V, simulations are performed using the proposed controller and the different navigation functions. Finally, the conclusions are in Section VI. To distinguish between matrices and geometric multivectors, all matrices are defined with a hat (e.g. A matrix \( \hat{A} \) and a multivector \( A \)).

II. MODEL

The following design is based on the camera-in-hand model [10], but can easily be extended to the camera-to-hand model and assumes the object being tracked is a sphere with a radius of \( \rho \), that contains a single visible point \( b \) and a unit vector \( a \) attached to the feature point. The position of this point and the direction of the unit vector are not important, but for convenience are chosen to be \( b = -\rho \vec{r}_3 \) and \( a = \vec{r}_2 \). The choice of target is not restrictive as a set of multiple isolated feature points can be modeled as if it were a sphere and arrow [1]. The global invertible transformation called the pinhole camera model is used to find the relationship between the Euclidean coordinates and the projected pixel coordinates.

\[
\pi_p : (\mathbb{R}^3 - \{0\}) \to \mathbb{R}^2
\]

\[
: p \to \hat{A}p
\]
where \( p \in \mathbb{R}^3 \) is a point and \( \hat{A} \in \mathbb{R}^{3 \times 3} \) is a constant upper triangle and invertible intrinsic camera-calibration matrix. In the control design it is assumed that the intrinsic camera matrix \( \hat{A} \) is well conditioned and invertible. Similar to the pinhole camera model system, we use a panoramic spherical camera with the following mapping

\[
\pi_s : (\mathbb{R}^3 - \{0\}) \to \mathbb{S}^2
\]

\[
\pi_s : p \to \frac{p}{\|p\|}
\]

where \( \mathbb{S}^2 = \{ v \in \mathbb{R}^3 : v \cdot v = 1 \} \) and \( p \in \mathbb{R}^3 \) is a point. After the project pixel coordinates are calculated the coordinates can be converted to spherical coordinates. Figure 1 shows the projection of the body to the image sphere. The coordinate frame \( \mathcal{F} \) is attached to the optical centre of a camera viewing the object. The location of the object relative to the camera is given by the rigid body translation \( d \). Via the mapping \( \pi_s \) the object will form a disc on \( \mathbb{S}^2 \) with a radius of

\[
\lambda = \frac{\rho}{\|d\|}
\]

where \( \rho + \epsilon < \|d\| < \infty \) is the radius of the body and \( \epsilon \) is an arbitrary small constant. The centre of the circle on the image-sphere is defined as

\[
c = \frac{d}{\|d\|}
\]

Using (1 and 2) we can now calculate the translation

\[
d = \frac{\rho}{\lambda} c
\]

Given the visible point \( b \) on the sphere and the vector \( a \) tangent to \( b \) we define the vectors

\[
q_1 = \frac{b}{\|b\|} = \frac{d - \rho r_3}{\|d - \rho r_3\|}
\]

and

\[
q_2 = \frac{(\hat{I} - q_1 q_1^T)a}{\|\hat{I} - q_1 q_1^T\|}
\]

\[
= \frac{(\hat{I} - q_1 q_1^T) r_2}{\|\hat{I} - q_1 q_1^T\| r_2}
\]

where \( r_1, r_2, r_3 \in \mathbb{R}^3 \) are the basis for the coordinate frame of the body \( \mathcal{F} \) and \( \hat{I} \) is the identity matrix. Using (3 and 4) we can calculate the orientation of the body \( \hat{R} \in SO(3) = [ \hat{R}_1 \ \hat{R}_2 \ \hat{R}_3 ] \).

\[
\hat{R}_3 = \frac{1}{\lambda} \left( c - (\cos(\phi) - \gamma) q_1 \right)
\]

\[
\hat{R}_2 = \frac{q_2 - \hat{R}_3 \cdot q_2}{\hat{R}_3 \cdot q_1}
\]

\[
\hat{R}_1 = \frac{q_2 - \hat{R}_3 \cdot q_2}{\hat{R}_3 \cdot q_1}
\]

where

\[
\cos(\phi) = q_1 \cdot c
\]

and

\[
\gamma = \sqrt{\lambda^2 - \sin^2(\phi)}
\]

Lastly, \( \hat{R}_1 \) is a normal to the plane formed by \( \hat{R}_2 \) and \( \hat{R}_3 \)

\[
\hat{R}_1 = \hat{R}_2 \times \hat{R}_3
\]

Given \( d \) and \( \hat{R} \) we can now calculate the transformation matrix

\[
\hat{H} = \begin{bmatrix} \hat{R}_1 & \hat{R}_2 & \hat{R}_3 & d \end{bmatrix}
\]

III. GEOMETRIC ALGEBRA

Geometric Algebra (GA) [11–14] is closely related to Clifford algebra [15] and it unifies many concepts of different special algebras. It includes many important concepts such as duality within projective geometry, Lie algebras and Lie groups, Plücker representations of lines, complex numbers, quaternions and dual quaternions. A GA, \( \mathcal{G}_{p,q} \) is defined by a geometric product between two multivectors \( A, B \in \mathbb{V}^n \) where \( \mathbb{V}^n \) is a linear space of dimension \( n = 2^{(p+q)} \). The geometric product is distributive, associative under addition and under multiplication by a scalar. The square of a vector is always a scalar. The canonical basis of \( \mathcal{G}_{p,q} \) is defined as the totally ordered set \( \{ e_1, \ldots, e_p, e_{p+1}, \ldots, e_{p+q} \} \) where \( e_i \) has the following properties:

\[
e_i e_j = \begin{cases} 
1 & \text{for } i = j \in \{1, 2, \ldots, p\} \\
-1 & \text{for } i = j \in \{p + 1, \ldots, p + q\} \\
0 & \text{for } i \neq j \in \{p + q + 1, \ldots, n\}
\end{cases}
\]

and

\[
e_{ij} = e_i \wedge e_j = -e_j \wedge e_i
\]

A. Geometric Product

The geometric product of two vectors \( a \) and \( b \) is defined as

\[
ab = \frac{1}{2} \langle ab + ba \rangle + \frac{1}{2} \langle ab - ba \rangle = a \cdot b + a \wedge b
\]

where \( \langle \cdot \rangle \) is called the inner product and \( \wedge \) is the outer product. In a space of dimension \( n \) there exist multivectors of grade 0 (scalars), grade 1 (vectors), grade 2 (bivectors), \( \ldots \), grade \( n \). An \( r \)-blade is a subspace structure that contains the outer product of \( r \) basis vectors, where \( r \) is the grade of the \( r \)-blade. The \( \langle M \rangle_r \) operator is defined as the grade-\( r \) entities of \( M \in \mathcal{G}_{p,q} \). The geometric product of two vectors \( a, b \in \langle \mathcal{G}_{p,q} \rangle_1 \) is equivalent to the inner product and the outer product of the vectors, which result in a scalar \( (a \cdot b) \in \langle \mathcal{G}_{p,q} \rangle_0 \) and a bivector \( (A \wedge B) \in \langle \mathcal{G}_{p,q} \rangle_2 \) respectively. A multivector \( M \) can be described as a linear combination of \( r \)-blades

\[
M = \sum_{i=0}^{r} \langle M \rangle_i
\]

Therefore the geometric product of two multivector \( A \) and \( B \) can be defined as

\[
AB = \sum_{i=0}^{r} \langle A \rangle_i \langle B \rangle_i = \sum_{j=0}^{s} \langle A \rangle_i \langle B \rangle_j
\]
where \( r, s \in \mathbb{R} \) are the maximum grade of \( A \) and \( B \) respectively. Thus, the inner product of two multivectors is
\[
A \cdot B \equiv \sum_{i=0}^{r} \langle A \rangle_i \cdot \langle B \rangle_j = \sum_{j=0}^{s} A \cdot \langle B \rangle_j
\]
and the outer product of two multivectors is
\[
A \wedge B \equiv \sum_{i=0}^{r} \langle A \rangle_i \wedge \langle B \rangle_j = \sum_{j=0}^{s} A \wedge \langle B \rangle_j
\]
where
\[
\langle A \rangle_i \cdot \langle B \rangle_j = \langle AB \rangle_{i-j}
\]
and
\[
\langle A \rangle_i \wedge \langle B \rangle_j = \langle AB \rangle_{i+j}
\]

B. Conformal Geometric Algebra

Conformal Geometric Algebra (CGA) [11]–[14], [16]–[21] is any Euclidean vector space \( \mathbb{R}^N \) extended by the Minkowski plane to generate null vectors. The Minkowski plane, \( \mathbb{G}_{1,1} \), has the orthonormal basis \( \{e_+, e_-\} \). A null basis can be defined as
\[
e_0 = \frac{1}{2}(e_- - e_+) \quad e_{\infty} = e_+ + e_-
\]
and then from (6) we define \( E \) as
\[
E = e_{\infty} \wedge e_0 = e_+ \wedge e_-
\]
The conformal vector space defined by using \( \mathbb{R}^3 \) is \( \mathbb{G}_{3+1,0,1} = \mathbb{G}_{4,1} \) and has the following basis \( \{1, e_-, e_+, e_1, e_2, e_3\} \). Given that the pseudo-scalar for Euclidean algebra is \( I_E = e_{123} \) the CGA pseudo-scaler for \( \mathbb{R}^3 \) is
\[
I_C = e_{+1-23} = EI_E
\]

C. Geometric Entities

One of the main advantages of CGA is its ability to represent different geometric entities. In \( \mathbb{G}_3 \) the basis entities are points and then other entities such as lines or planes are defined from points. In CGA, \( \mathbb{G}_{4,1} \), the basis entities are spheres and entities such as points are just degenerate spheres of zero radius. A point \( x \in \mathbb{G}_{4,1} \) satisfies the following properties
\[
x^2 = 0 \quad x \cdot e_{\infty} = -1
\]
The point satisfying these properties is defined as
\[
x = x + \frac{1}{2}x^2e_{\infty} + e_0
\]
where \( x \in \mathbb{G}^3 \). The Conformal Split can be used to reduce the dimension of \( x \).
\[
x = (x \wedge \mathbf{E})\mathbf{E}
\]
The dual of a sphere \( s^* \) can be found by using four points on the sphere.
\[
s^* = x_1 \wedge x_2 \wedge x_3 \wedge x_4
\]
where \( x_1, x_2, x_3, x_4 \in \mathbb{G}_{4,1} \) are points on the sphere. If a point \( x \) is on a sphere \( s \) (or any other object) it will fulfill the following property \( x \wedge s^* = 0 \) or \( x \cdot s = 0 \).

D. Euclidean Distance

The Euclidean distance \( \chi \) between two points \( x_1, x_2 \in \mathbb{G}_{4,1} \)
\[
\chi = \sqrt{-2(x_1 \cdot x_2)}
\]
\[
= \sqrt{\frac{2((x_1 + \frac{1}{2}x^2e_{\infty} + e_0) \cdot (x_2 + \frac{1}{2}x^2e_{\infty} + e_0))}{2}}
\]
\[
= \sqrt{\frac{2(x_1 \cdot x_2 - \frac{1}{2}x_1^2 - \frac{1}{2}x_2^2)}{2}}
\]
\[
= \sqrt{(o_{1x} - o_{2x})^2 + (o_{1y} - o_{2y})^2 + (o_{1z} - o_{2z})^2}
\]
where \( x_1 = (x_1 \wedge \mathbf{E})\mathbf{E} = [o_{1x}, o_{1y}, o_{1z}] \in \mathbb{G}_3 \) and \( x_2 = (x_2 \wedge \mathbf{E})\mathbf{E} = [o_{2x}, o_{2y}, o_{2z}] \in \mathbb{G}_3 \). Thus, given two spheres \( s_1, s_2 \in \mathbb{G}_{4,1} \) where
\[
s_i = c_i + \frac{1}{2}(c_i^2 - \rho_i^2)e_{\infty} + e_0 = c_i - \frac{1}{2}\rho_i^2e_{\infty}
\]
where \( c_i \in \mathbb{G}_{4,1} \) is the centre of the sphere, \( c_i = (c_i \wedge \mathbf{E})\mathbf{E} \in \mathbb{G}_3 \) and \( \rho_i \in \mathbb{R} \) is the radius of the sphere. The distance \( \chi \) between the two spheres is
\[
\chi = \sqrt{-2(c_1 \cdot c_2) - \rho_1 - \rho_2}
\]
Next, the distance between a plane \( P \in \mathbb{G}_{4,1} \) and a point \( x \in \mathbb{G}_{4,1} \) is
\[
\chi = P \cdot x
\]
\[
= (n + v e_\infty) \cdot (x + \frac{1}{2}x^2e_{\infty} + e_0)
\]
\[
= n \cdot x - v
\]
where \( n \in \mathbb{G}_3 \) is the unit normal to the plane and \( v \in \mathbb{R} \) is the distance from the plane to the origin. Therefore, the distance between a plane and a sphere is
\[
\chi = P \cdot c - \rho
\]
where \( c \in \mathbb{G}_{4,1} \) is the centre of the of the sphere and \( \rho \in \mathbb{R} \) is the radius of the sphere.

E. Rigid Body Motion

In CGA, rigid body motion is a time dependent multivector called a motor which is comprised of a rotation and a translation. The rotation of a point \( x_0 \) is defined as
\[
x(t) = \mathbf{R}(t)x_0\mathbf{R}^{-1}(t)
\]
where
\[
\mathbf{R}(t) = \exp \left( \frac{\theta(t)}{2} \mathbf{L} \right)
\]
\[
= \cos \left( \frac{1}{2}\theta(t) \right) + L\sin \left( \frac{1}{2}\theta(t) \right)
\]
where \( L \in \langle \mathbb{G}_3 \rangle_2 \) is a unit bivector that represents the dual form of the axis of rotation and \( \theta(t) \in \mathbb{R} \) is the angle of rotation. The translation of a point \( x_0 \) is defined as
\[
x(t) = T(t)x_0T^{-1}(t)
\]
where
\[
T(t) = 1 + \frac{1}{2}e_{\infty}n(t) = \exp \left( \frac{1}{2}e_{\infty}n(t) \right)
\]
where \( \mathbf{n} \in (\mathbb{G}_3)_1 \) is a translation vector. Then, if we apply a rotor and a translator together we have a motor which is defined as

\[
M(t) = T(t)R(t)
\]

and

\[
x(t) = M(t)x_0M^{-1}(t)
\]

**F. Kinematics**

Using (9) and (10) to differentiate (11) we obtain

\[
\dot{M}(t) = \frac{1}{2}V(t)M(t) \text{ and } \dot{M}^{-1}(t) = -\frac{1}{2}V(t)M(t)
\]

where \( V(t) \) is the velocity bivector of \( M(t) \). The time derivatives of (9) and (10) are

\[
\begin{align*}
\dot{R}(t) &= \frac{L}{2} \theta(t) \exp \left( \frac{\theta(t)}{2} L \right) = \frac{1}{2} \mathbf{w}(t)I_E R(t) \\
\dot{T}(t) &= \frac{1}{2} e_\infty \hat{\mathbf{n}}(t) \exp \left( \frac{1}{2} e_\infty \hat{\mathbf{n}}(t) \right) = \frac{1}{2} e_\infty \hat{\mathbf{n}}(t) T(t)
\end{align*}
\]

where \( \mathbf{w}(t) \in \mathbb{G}_3 \) is the rotational velocity of the object about \( L \) and \( e_\infty \hat{\mathbf{n}} = e_\infty \hat{\mathbf{x}} \) is the velocity \( \mathbf{v}(t) \) assuming that the initial rotation is 1. Taking the derivative of (12) and using (13) we obtain

\[
\begin{align*}
\dot{x}(t) &= \dot{M}(t)x_0M^{-1}(t) + M(t)x_0\dot{M}^{-1}(t) \\
&= \left( \frac{1}{2} V(t)M(t) \right) x_0M^{-1}(t) \\
&\quad + M(t)x_0 \left( -\frac{1}{2} M^{-1}(t)V(t) \right) \\
&= \frac{1}{2} V(t)x(t) - \frac{1}{2} \dot{x}(t)V(t) \\
&= V(t) \cdot x(t)
\end{align*}
\]

Decomposing the motor in (11) into a translation followed by a rotation and taking the derivative we obtain

\[
\begin{align*}
\dot{M}(t) &= \dot{R}(t)T(t) + R(t)\dot{T}(t) \\
&= \frac{1}{2} \mathbf{w}(t)I_E R(t)T(t) \\
&\quad + \frac{1}{2} R(t)e_\infty \hat{x}(t)[R^{-1}(t)R(t)]T(t) \\
&= \frac{1}{2} (\mathbf{w}(t)I_E + R(t)e_\infty \hat{x}(t)R^{-1}(t))R(t)T(t) \\
&= \frac{1}{2} (\mathbf{w}(t)I_E + R(t)e_\infty \hat{x}(t)R^{-1}(t))M(t)
\end{align*}
\]

From equation (13) and (14) we derive the screw velocity vector \( \mathbf{V}(t) \)

\[
\begin{align*}
\mathbf{V}(t) &= \mathbf{w}(t)I_E + e_\infty \hat{x}(t)R^{-1}(t) \\
&= \mathbf{w}(t)I_E + e_\infty \mathbf{v}(t)
\end{align*}
\]

where \( \mathbf{v}(t) = R(t)\hat{x}R^{-1}(t) \) or when the initial rotation is 1 then \( \mathbf{v}(t) = \hat{x}(t) \). With a shift in base equation (14) becomes

\[
\begin{align*}
\mathbf{V'}(t) &= \mathbf{w}(t)I_E + e_\infty \mathbf{v}'(t) \\
&= \mathbf{w}(t)I_E + e_\infty (\mathbf{v}(t) + \mathbf{w}(t) \wedge \mathbf{r}(t))
\end{align*}
\]

where \( \mathbf{r}(t) \in \mathbb{G}_3 \) is the shift in base.

**IV. Control**

The control objective is to move a camera to a desired position and orientation without the object leaving the image or the camera colliding with another object. It is assumed that the linear and angular velocities of the camera can be independently controlled, that the camera calibration matrix is known and the camera is modeled as a sphere. This means that in Euclidean space \( \dot{R}(t) \rightarrow \hat{I} \) and \( \|\mathbf{n}(t)\| \to 0 \) as \( t \to \infty \) or in CGA \( R(t) \to 1, T(t) \to 1, M(t) \to 1 \) and \( V(t) \to 0 \) as \( t \to \infty \). The transformation matrix from (5) can be converted into an error motor using (9, 10 and 11) to represent both the current and desired orientation in motors. The error motor is defined as

\[
M_e(t) = M_d(t)M^{-1}(t)
\]

where \( M_d(t) \) is the desired motor and \( M(t) \) is the current motor. Given that a rotor can be extracted from the motor by using the conformal split

\[
R(t) = (M(t) \wedge \mathbf{E})\mathbf{E}
\]

The error in rotation is defined as

\[
R_e(t) = (M_e(t) \wedge \mathbf{E})\mathbf{E}
\]

(14)

Lastly, the error in translation is defined as

\[
T_e(t) = R_e^{-1}(t)M_e(t)
\]

From (14) we define an angular velocity controller as

\[
\mathbf{w}(t) = \hat{K}_w(R_e(t))_2I_E
\]

(15)

where \( \mathbf{w}(t) \in \mathbb{G}_3 \) is the angular velocity of the camera and \( \hat{K}_w \) is a positive constant gain matrix. The linear velocity controller is defined as

\[
\mathbf{v}(t) = \hat{K}_v(T_e(t))_2 \cdot \mathbf{e}_0
\]

(16)

where \( \mathbf{v}(t) \in \mathbb{G}_3 \) is the linear velocity and \( \hat{K}_v \) is a positive constant gain matrix. From (14) when there is a shift in base (16) becomes

\[
\mathbf{v'}(t) = \mathbf{v}(t) + \mathbf{w}(t) \wedge \mathbf{r}(t)
\]
A. Navigation Functions

Navigation functions [22] can be used to ensure that the object stays in the edges of the image. Four repulsive planes are formed around the edges of the image as shown in Figure 2. Using (8) to calculate the distance $\chi_i$ between the sphere and the plane $P_i$ we redefine (16) as

$$v(t) = \hat{K}_v(T_v(t)T_{nf})_2 \cdot e_0$$  \hspace{1cm} (17)

where

$$T_{nf} = \exp(-K_i \sum_{i=1}^{4} \chi_i)$$

where $\chi_i \in \mathbb{R}$ is the minimum safe distance the camera should be to the object to avoid, $c \in \mathbb{G}_{4,1}$ is the camera and

$$K_i = \begin{cases} -ke_2 & \text{if } i = 1 \text{ and } \chi \leq \zeta \\ -ke_1 & \text{if } i = 2 \text{ and } \chi \leq \zeta \\ ke_1 & \text{if } i = 3 \text{ and } \chi \leq \zeta \\ ke_2 & \text{if } i = 4 \text{ and } \chi \leq \zeta \\ 0 & \text{if } \chi > \zeta \end{cases}$$

and $k \in \mathbb{R}$ is a positive constant gain. The planes on the edge of the image are defined by the points in the corners of the image and the origin (the pinhole) of the camera.

$$x_1 = \hat{A}^{-1} \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T$$
$$x_2 = \hat{A}^{-1} \begin{bmatrix} u_{max} & 0 & 1 \end{bmatrix}^T$$
$$x_3 = \hat{A}^{-1} \begin{bmatrix} 0 & v_{max} & 1 \end{bmatrix}^T$$
$$x_4 = \hat{A}^{-1} \begin{bmatrix} u_{max} & v_{max} & 1 \end{bmatrix}^T$$

Once three points on each plane are known, the planes can easily be defined as

$$P_1 = (e_\infty \wedge e_0 \wedge x_2 \wedge x_1)I_C$$
$$P_2 = (e_\infty \wedge e_0 \wedge x_1 \wedge x_3)I_C$$
$$P_3 = (e_\infty \wedge e_0 \wedge x_4 \wedge x_2)I_C$$
$$P_4 = (e_\infty \wedge e_0 \wedge x_3 \wedge x_4)I_C$$

For the camera to avoid an obstacle we model the camera and the obstacle as spheres and then use (7) to calculate the distance between the two spheres. If the two spheres are within a small distance $\Delta$ the camera will change velocity to a tangent to the obstacle.

$$v'(t) = \begin{cases} v(t) & \text{if } \chi > \Delta \\ v_{tan}(t) & \text{otherwise} \end{cases}$$

where

$$v_{tan}(t) = v(t) \wedge v_2$$

$$v_2 = \begin{cases} c_1 - c_2 & \text{if } c_1 \cdot c_2 > \delta \\ e_1r_1 + e_2r_2 + e_3r_3 & \text{otherwise} \end{cases}$$

where $\delta \in \mathbb{R}$ is a small positive constant, $-1 \leq r_i \leq 1 \in \mathbb{R}$ is a random number to avoid the camera getting stuck if it is heading directly towards the object and $c_1$ and $c_2$ are the centre of the camera sphere and the centre of the obstacle sphere.

B. Stability

The controller in (15) and (17) ensures asymptotic translation and rotation regulation, i.e. $\| \dot{M}(t) \| \to 1$, $\| \dot{R}(t) \| \to 1$, $\| T(t) \| \to 1$, $\| V(t) \| \to 0$ as $t \to \infty$. Let $T(t) \in \mathbb{G}_{4,1}$ denote the following non-negative positive definite function (i.e. a Lyapunov candidate):

$$L(t) = (1 - \langle R_0 \rangle)^2 - \langle R_2 \rangle^2 + \langle T \rangle_0 + \langle V \rangle_2 \cdot e_0$$  \hspace{1cm} (18)

Given that $R_e = R = \langle R_0 \rangle + \langle R \rangle_2$ and $\dot{R} = \langle \dot{R}_0 \rangle + \langle \dot{R} \rangle_2$ from (9) we derive

$$\langle \dot{R}_0 \rangle = -\frac{1}{2} \sin \left( \frac{1}{2} \theta \right) \theta$$
$$\langle \dot{R}_2 \rangle = -\frac{w}{2T} \langle R_2 \rangle$$
and

$$\langle \dot{T} \rangle_2 = \frac{1}{2} \dot{\theta} I_R = \frac{1}{2} w$$

Similarly, given that $T, T_{nf} = T = \langle T_0 \rangle + \langle T \rangle_2$ and $\dot{T} = \langle \dot{T}_0 \rangle + \langle \dot{T} \rangle_2$ from (10) we derive

$$\langle \dot{T}_0 \rangle = 0$$

and

$$\langle \dot{T} \rangle_2 = \frac{1}{2}(\dot{v} - \sum_{i=1}^{4} \chi_i) e_\infty = \frac{1}{2}(v - \sum_{i=1}^{4} v_{nf,i})$$  \hspace{1cm} (22)

Thus, from (19-22) the time derivative of $L(t)$ can be determined as follows

$$\dot{L} = -2(1 - \langle R_0 \rangle) \langle \dot{R}_0 \rangle - 2\langle R_2 \rangle \langle \dot{R}_2 \rangle_2 + 2(\langle T \rangle_2 \cdot e_0)(\langle \dot{T} \rangle_2 \cdot e_0)$$
$$= \frac{1}{2T}(1 - \langle R_0 \rangle)\langle R_2 \rangle_2 w - \langle R_2 \rangle_2 w + (\langle T \rangle_2 \cdot e_0)(v - \sum_{i=1}^{4} v_{nf,i})$$
$$= \frac{1}{2T}(1 - \langle R_0 \rangle)\langle R_2 \rangle_2 K_w \langle R_2 \rangle_2$$
and

$$\langle \dot{T} \rangle_2 = \frac{1}{2}(\langle R_2 \rangle_2 K_w \langle R_2 \rangle_2 \cdot e_0)(\langle \dot{T} \rangle_2 \cdot e_0))$$

Where the initial target sphere is inside of the image, i.e. $P_1 \cdot \rho_i \cdot \rho > 0$, $i = \{1, 2, 3, 4\}$ and the velocity of the camera is never too fast that the camera loses the target, i.e $P_1 \cdot McM^{-1} > \rho_i \cdot \rho > 0$, $i = \{1, 2, 3, 4\}$. For the point $b$ to be visible $(d - \rho_3) \cdot \rho_3 > 0$. From (18) and (23) and by Lyapunov stability the controllers in (15) and (17) are asymptotically stable.

V. EXPERIMENTAL RESULTS

To validate the new controller we performed a series of simulations. The experimental setup included a robot with six DOF and a camera mounted on its end effector that can be contained within a 1m sphere. The camera calibration matrix used was $A = \begin{bmatrix} 100 & 0 & 320 \\ 0 & 100 & 240 \\ 0 & 0 & 1 \end{bmatrix}$ and the constant gain matrices used in the controller were $K_v = 0.5 I$ and $K_w = 0.2 I$. For simplicity every iteration of the simulation was 1s.
Initially, a desired image was calculated and the robot had to translate and rotate to re-create the desired image (teachby-showing [10]). In Figure 3-4 the controller was given a desired position outside of the image and the controller had to keep the object in the image. In Figure 5 the camera had to avoid obstacles. In all of the figures the green sphere is the desired position, the blue sphere is the final position and the red spheres are the trajectory. The black spheres are the obstacles and the black lines are the boundary of the image.

VI. CONCLUSION

The combination of CGA and geometric visual servoing using spheres provides many advantages to visual servoing problems. One such advantage was the ease at which navigation functions can be designed in CGA and added to the controller. CGA can easily handle the different combinations of spheres, points and planes necessary in the algorithm. Also no additional conversions were needed between $\mathbb{R}^3$ and $\mathbb{G}_{4,1}$ as the navigation functions were also implemented in CGA.

Future works include to fully implement the controller on physical robots.

REFERENCES