Robust Dynamic Control of an Arm of a Humanoid using Super Twisting Algorithm and Conformal Geometric Algebra

O. Carbajal-Espinosa[1], L. González-Jiménez[2], IEEE member A. Loukianov[3] and IEEE senior member E. Bayro-Corrochano[4]

Abstract The pose tracking problem for the 5 DOF (degrees of freedom) arm of a humanoid robot is studied. The kinematic and dynamic models of the manipulator are obtained using the conformal geometric algebra framework. Then, using the obtained models, the well known super-twisting algorithm, is used to design a controller in terms of the conformal geometric algebra for the pose tracking problem. Simulation shows the performance of the proposed controller with the conformal models for the tracking an object.

Key words: Kinematics modeling, Dynamic modeling, Super Twisting Algorithm, Conformal Geometric Algebra, Humanoid manipulator

1 Introduction

The geometrical relationships between the kinematical chains of a humanoid and an object of interest in the environment, determine the reference for their position and orientation. To follow a target on the task space, a controller must be designed in order to assure a proper torque values on each joint of the manipulator. First, models of the kinematics and dynamics of the robot must be obtained, for this end, the conformal geometric algebra (CGA) framework allows the representation of rigid transformations (rotations, translations, screw motions) and geometric entities (points, lines, spheres, etc), these entities will serve to model the structure of the robot, for example, lines can be considered as links of the manipulator. Moreover, the composition

[1],[3],[4] are with CINVESTAV,
Department of Electrical Engineering and Computer Sciences Unidad Guadalajara
e-mail: ocarbajal[1],louk[3],edb[4]@gdl.cinvestav.mx
[2] is with ITESO,
Department of Electronic, Systems and Informatics
e-mail: luisgonzalez@iteso.mx

of several rigid transformations acting over a geometric entity can be computed as a sequence of geometric products of consecutive motors (the conformal entity that represents a 3D rigid transformation) [1, 2]. In this work, the conformal geometric algebra (CGA) framework is used to obtain the direct, differential kinematical and forward dynamic models of the arm of a humanoid. The rest of the work is organized as follows. Section II presents an introduction to Conformal Geometric Algebra framework, introducing some geometric primitives and rigid transformations in CGA used in this work. Sections III presents the kinematic and dynamic modeling of serial manipulator. In Section IV and V present the control strategy and the simulation results of the proposed method. Finally, some conclusions are given in Section VI.

2 Geometric Algebra

We will use G_n to denote the geometric algebra of n-dimensions, which is a graded-linear space. As well as vector-addition and scalar multiplication, we have a non commutative product which is associative and distributive over addition. This is the *geometric* or *Clifford product*. The inner product of two vectors is the standard *scalar* or *dot* product, which produces a scalar. The outer or wedge product of two vectors is a new quantity which we call a *bivector*.

In this paper we will specify the geometric algebra G_n by $G_{p,q,r}$, where p, q and r stand for the number of basis vectors which square to 1, -1 and 0 respectively and fulfill $n = p+q+r$. The entire basis of G_n is defined as the ordered set:

$$\{1\},\{e_i\},\{e_i \wedge e_j\},\{e_i \wedge e_j \wedge e_k\},\ldots,\{e_1 \wedge e_2 \wedge \ldots \wedge e_n\} \tag{1}$$

Where e_i denote the basis vector i, and has the following properties:

$$e_i e_j = \begin{cases} 1 \ for\ i = j \in 1,\cdots,p \\ -1 \ for\ i = j \in p+1,\cdots,p+q \\ 0 \ for\ i = j \in p+q+1,\cdots,p+q+r. \\ e_i \wedge e_j \ for\ i \neq j \end{cases}$$

The *Conformal Geometric Algebra*, $G_{4,1} = G_{4,1,0}$, can be used to treat conformal geometry in a very elegant way, representing the Euclidean vector space $\mathbb{R}^3$ in $\mathbb{R}^{4,1}$, for a more complete treatment, the reader is referred to the texts by [1, 2]. This space has an orthonormal vector basis given by $\{e_i\}$ and $e_{ij} = e_i \wedge e_j$ are bivectorial bases. The unit Euclidean pseudo-scalar $I_e := e_1 \wedge e_2 \wedge e_3$, a pseudo-scalar $I = I_e E$, and the bivector $E := e_4 \wedge e_5 = e_4 e_5$ are used for computing Euclidean and conformal duals of multivectors. A null can be defined as

$$e_\infty = e_4 + e_5, \quad e_0 = \frac{1}{2}(e_4 - e_5) \tag{2}$$

where e_∞ is the point at infinity and e_0 is the origin point. These two null vector satisfies

$$e_\infty^2 = e_0^2 = 0, \;\; e_\infty \cdot e_0 = 1$$

The *point* is written as

$$x_c = x_e + \frac{1}{2}x_e^2 e_\infty + e_0$$

Given two conformal points x_c and y_c, it can be defined $x_c - y_c = (y_c \wedge x_c) \cdot e_\infty$ and, consequently, the following equality is fulfilled as well:

$$(x_c \wedge y_c + y_c \wedge z_c) \cdot e_\infty = (x_c \wedge z_c) \cdot e_\infty \tag{3}$$

The point will be use to define the position of each joint and the center of mass of each link on a manipulator.

The *Lines* will be used to define the rotation axes and orientation of the manipulator, and can be defined in CGA as a circle passing through the point at infinity. The OPNS (Outer Product Null Space) form of a line is represented as

$$L^* = x_{c_1} \wedge x_{c_2} \wedge e_\infty \tag{4}$$

The standard IPNS (Inner Product Null Space) form of the line can be expressed as

$$L = \mathbf{n}I_e - e_\infty \mathbf{m}I_e \tag{5}$$

where **n** and **m** stand for the line orientation and moment respectively. Given two lines L_a and L_b we can define a third error line L_e as

$$L_\varepsilon = L_a - L_b \tag{6}$$

Rigid transformations can be expressed in conformal geometric algebra by carrying out successive plane reflections.

The *translation* to a vector $a \in \mathbb{R}^3$, of conformal geometric entities can be done by carrying out two reflections in parallel planes π_1 and π_2. That is $T_a Q \widetilde{T_a}$, where

$$T_a = (n + de_\infty)n = 1 + \frac{1}{2}ae_\infty = e^{\frac{a}{2}e_\infty} \tag{7}$$

The *rotation* is the product of two reflections at nonparallel planes which pass through the origin. That is $R_\theta Q \widetilde{R_\theta}$, where

$$R_\theta = cos(\frac{\theta}{2}) - sin(\frac{\theta}{2})l = e^{-\frac{\theta}{2}l} \tag{8}$$

with $l = n_2 \wedge n_1$, and θ twice the angle between the planes π_2 and π_1.

The screw motion *called motor* related to an arbitrary axis L is $M = TR\widetilde{T}$ and it is applied in the same way than a rotor, $M_\theta Q \widetilde{M_\theta}$, where

$$M_\theta = TR\widetilde{T} = cos(\frac{\theta}{2}) - sin(\frac{\theta}{2})L = e^{-\frac{\theta}{2}L} \tag{9}$$

3 Kinematic and Dynamic Modeling Of Manipulators

The *direct kinematics* of a manipulator consists in calculating the position and orientation of the end-effector of a serial robot using the values of the joint variables. If the joint variable is a translation, $M_i = T_i = \exp^{-dne_\infty}$ for a prismatic joint and a rotation $M_i = R_i = \exp^{\frac{-\theta L_r}{2}}$ for a revolutive joint. The direct kinematics for a serial robot is a successive multiplication of motors given by

$$Q' = M(q)_1 \dots M(q)_n Q \widetilde{M(q)}_n \dots \widetilde{M(q)}_1 = \left(\prod_{i=1}^{n} M(q)_i Q \prod_{i=1}^{n} \widetilde{M(q)}_{i=n-i+1} \right) \tag{10}$$

for a given angular or translation position vector $q = [q_1 \ \dots \ q_n]^T$.

The *differential kinematics* of the system results from the differentiation of (10) for points and lines, and is given by

$$\dot{x}'_p = J_x \dot{q}, \quad \dot{L}'_p = J_L \dot{q} \tag{11}$$

with $\dot{q} = [\dot{q}_1 \ \dots \ \dot{q}_n]$, and the Jacobian matrices defined as $J_x = \left[x'_p \cdot L'_1 \ \dots \ x'_p \cdot L'_n \right]$, $J_L = [\alpha_1 \ \dots \ \alpha n]$, where

$$L'_j = \left(\prod_{i=1}^{j-1} M_i \right) L_j \left(\prod_{i=1}^{j-1} \widetilde{M}_{j-i} \right), \quad \alpha_j = \frac{1}{2} \left(L'_p L'_j - L'_j L'_p \right) \tag{12}$$

and L_i is the axis for the i^{th} joint in the initial position. Please refer to [3] for a more detailed explanation about the differentiation process.

On the other hand, we can write the *dynamic equations* of the system using the Euler-Lagrange equations [11] as

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q) = \tau \tag{13}$$

It is possible rewrite this equation in the geometric algebra framework [12], defining the matrix $M(q) = M_v + M_I$, where M_v and M_I are defined as follow. The matrix M_I can be written as the product of two matrix δ and I if we define them as

$$M_I = \delta I = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ 0 & 1 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{pmatrix} \begin{pmatrix} I_1 & 0 & \cdots & 0 \\ I_2 & I_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ I_n & I_n & \cdots & I_n \end{pmatrix}. \tag{14}$$

where I_n are the inertial value of each link. The matrix M_v also can be expressed as the product of two matrices $M_v = V^T m V$, where the matrices V and m are

$$m := \begin{pmatrix} m_1 & 0 & \cdots & 0 \\ 0 & m_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & m_n \end{pmatrix}, \quad V := \begin{pmatrix} x'_{cm1} \cdot L'_1 & 0 & \cdots & 0 \\ x'_{cm2} \cdot L'_1 & x'_{cm2} \cdot L'_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ x'_{cmn} \cdot L'_1 & x'_{cmn} \cdot L'_2 & \cdots & x'_{cmn} \cdot L'_n \end{pmatrix}. \tag{15}$$

where the values m_i are the mass of each link and x'_{cmi} and L'_i are the center of mass and the rotation axes of each link obtained by the direct kinematic defined in (10) and (12) respectively. Based in the properties of the matrices M and C, it can be written the matrix C without derivatives as $C = V^T m\dot{V}$, where the matrix $\dot{V}$ can be obtained following the next mathematical procedure. Defining $V = XL$ yields

$$V = \begin{pmatrix} x'_{cm1} & 0 & \cdots & 0 \\ 0 & x'_{cm2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & x'_{cmn} \end{pmatrix} \begin{pmatrix} L'_1 & 0 & \cdots & 0 \\ L'_1 & L'_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ L'_1 & L'_2 & \cdots & L'_n \end{pmatrix}, \tag{16}$$

then $\dot{V} = \dot{X}L + X\dot{L}$. With $\dot{X} = diag[\dot{x}'_{cm1}, \cdots, \dot{x}'_{cmn}]$ an diagonal matrix, where the $\dot{x}'_{cmi}$ is computed using the differential kinematic (11).

On the other hand

$$\dot{L}' = \begin{pmatrix} \dot{L}'_1 & 0 & \cdots & 0 \\ \dot{L}'_1 & \dot{L}'_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \dot{L}'_1 & \dot{L}'_2 & \cdots & \dot{L}'_n \end{pmatrix}. \tag{17}$$

where $\dot{L}'_i$ is computed using (12) and can be expressed in a matrix form as:

$$\begin{pmatrix} \dot{L}'_1 \\ \dot{L}'_2 \\ \vdots \\ \dot{L}'_n \end{pmatrix} = \frac{1}{2} \left[\begin{pmatrix} L'_1L'_1 & 0 & \cdots & 0 \\ L'_2L'_1 & L'_2L'_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ L'_nL'_1 & L'_nL'_2 & \cdots & L'_nL'_n \end{pmatrix} - \begin{pmatrix} L'_1L'_1 & 0 & \cdots & 0 \\ L'_1L'_2 & L'_2L'_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ L'_1L'_n & L'_2L'_n & \cdots & L'_nL'_n \end{pmatrix} \right] \dot{q}. \tag{18}$$

Finally the vector g is expressed as the product of the three matrices as $G = V^T ma$, where V have been defined and the vector $F = ma$ is a force component, where $m = diag[m_1, m_2, \cdots, m_n]$ is a $n \times n$ matrix of masses and $a = [ge_2, ge_2, \cdots]^T$ is a $n \times 1$ vector.

4 Dynamic Control

Now using (11) and (13) we will define the output tracking problem for the position x'_{p1} and the orientation L'_{p1} of the manipulator of the humanoid robot. A state-space model can be obtained using the next state variables: $x_1 = Pose\ of\ the\ manipulator$, $x_2 = q$ and $x_3 = \dot{q}$. Using the state variables, (11) and (13) the state-space model for the pose of the manipulator can be defined as:

$$\begin{aligned} \dot{x}_1 &= Jx_3 \\ \dot{x}_2 &= x_3 \\ \dot{x}_3 &= -M^{-1}Cx_3 - M^{-1}G + M^{-1}\tau \end{aligned} \tag{19}$$

where J is the Jacobian matrix, $-M^{-1}Cx_3 - M^{-1}G$ will be expressed by $f(x_2, x_3)$ and $y = x_1$ is the output of the system.

Now, we define a reference signal $x_{ref1}(t)$ for the arm and we will define an error signal. Omitting the parenthesis of the reference, the tracking error is given by $\varepsilon_1 = x_1 - x_{ref1}$. Assuming that we know the derivative $\dot{x}_{ref1}$ and the second derivative $\ddot{x}_{ref1}$, the derivative of the error gives

$$\dot{\varepsilon}_1 = Jx_3 - \dot{x}_{ref1} \tag{20}$$

From (20) we will define a virtual control variable, which ensure that the variable ε_1 will tends to zero. This new variable is obtained as in block control like

$$x_{ref3} = J^{-1}\left(-k_1\varepsilon_1 + \dot{x}_{ref1}\right) \tag{21}$$

Now we will define an error variable for the second block as $\varepsilon_2 = x_3 - x_{ref3}$. Immediately, the error signal ε_2 is differentiated and gives

$$\dot{\varepsilon}_2 = f(x_2, x_3) + M^{-1}\tau - \dot{x}_{ref3} \tag{22}$$

the signal $\dot{x}_{ref3}$ its divided in two terms, one known part (δ_J)

$$\delta_J = J^{-1}(-k_1(Jx_3 - \dot{x}_{ref1}) + \ddot{x}_{ref1}) \tag{23}$$

and an unknown part Δ_J. On this way the derivative of ε_s can be rewritten as

$$\dot{\varepsilon}_2 = f(x_2, x_3) + M^{-1}\tau - \delta_J - \Delta_J \tag{24}$$

Now using (24), we are able to design a control law for τ, to tackle the pose tracking problem. For this end we will use a **Super-Twisting Algorithm** applied to the robotic manipulator. We will design our control law τ as

$$\tau = M(u_0 + u_1) \tag{25}$$

where the term u_0 is used to reject the nominal part of the equation (24) and the known signals, and is defined as $u_0 = -f(x_2, x_3) + \delta_J$, this term ensure the sliding mode occurrence from initial instance [7]. Now we use the sliding surface $s = \varepsilon_2$. The derivative of s using u and the control term u_0 yields $\dot{s} = u_1 - \Delta_J$.

To induce an sliding mode in $\dot{s}$ we design the second part u_1 of the control law using the super-twisting [9, 10] algorithm as

$$\begin{aligned} u_1 &= -\sigma N sign(s) + \mu \\ \dot{\mu} &= \Sigma sign(s) \end{aligned} \tag{26}$$

where $N = |s|^{\rho}$ and σ, ρ, Σ are design parameters. The stability proof and the system convergence, is demonstrated in [9], for more detail please refer to the cited reference.

5 Simulation Results

The initial values of the rotation axes, the values of x_i and the center of mass x_{cmi}, for $i = 1 \cdots 5$, in centimeters, used in the simulation are

$$
\begin{aligned}
&L_1 = e_{23} + e_\infty (x_1 \cdot e_{23}), \qquad x_1 = 25e_2, \qquad x_{cm1} = -8.91e_1 + 24.92e_2 \\
&L_2 = e_{31} + e_\infty (x_2 \cdot e_{31}), \; x_2 = -15.3e_1 + 25e_2, \; x_{cm2} = -15.23e_1 + 25e_2 + 0.1e_3 \\
&L_3 = e_{23} + e_\infty (x_3 \cdot e_{23}), \; x_3 = -15.3e_1 + 25e_2, \; x_{cm3} = -21.8e_1 + 25.1e_2 \\
&L_4 = e_{31} + e_\infty (x_4 \cdot e_{31}), \; x_4 = -30.5e_1 + 25e_2, \; x_{cm4} = -31.08e_1 + 25.23e_2 + 0.79e_3 \\
&L_5 = e_{32} + e_\infty (x_5 \cdot e_{32}), \; x_5 = -30.5e_1 + 25e_2, \; x_{cm5} = -41.89e_1 + 25.22e_2
\end{aligned} \tag{27}
$$

The euclidean component of the references signals are given by $L_{ref} = [0, 1, -1]^T$ and $x_{ref} = [7cos(2t), 24, 14 + 5sin(2t)]^T$, where x_{ref} and L_{ref} are the references for the position and orientation, respectively. The mass of the links was $m_1 = 0.45kg$, $m_2 = 0.05kg$, $m_3 = 0.34kg$, $m_4 = 0.1kg$, $m_5 = 0.11kg$. The values of the controller was $k_1 = [7,\ 7,\ 5,\ 21,\ 22,\ 24]^T$, $\sigma = 80$, $\rho = 0.5$ and $\Sigma = 6$.

The figure (1 a,b) shows the tracking response of the arm, the figure (1 c) depicts the error signals for the pose of the arm. Finally the figure (2) shows a sequence of the humanoid torso using the angular values of the joints obtained with the proposed method in a virtual model developed in CLUCalc [6] and Matlab [5].

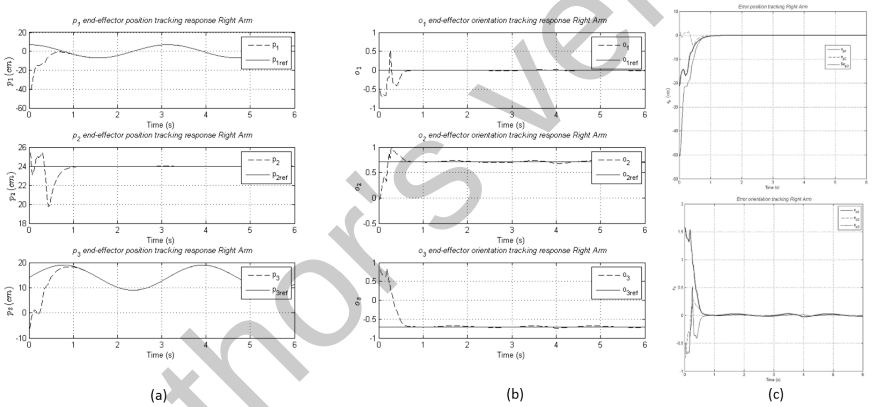

Fig. 1 (a) Position of the end-effector. (b) Orientation of the end-effector. (c) Error signal.

6 Conclusions

The conformal geometric algebra was used to define, in a simple and compact manner, the kinematic and dynamic models of the arm of a humanoid. Furthermore, a super-twisting controller was designed in this framework which allows to define

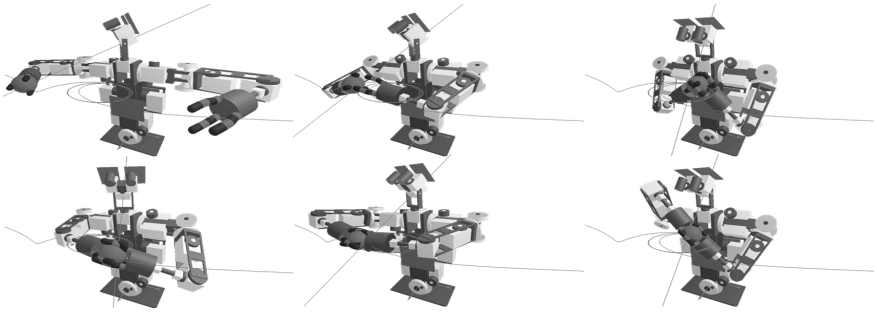

Fig. 2 Image sequence of the simulation results.

error variables between complex geometric entities. The proposed control scheme is robust against external disturbances, parameter variations and model uncertainties. Moreover, using a super twisting controller results in chattering-free control signals and finite time convergence of the closed loop system. This work pursues the inclusion of geometric restrictions expressed in CGA into the model of robotic systems, and in the controller design procedure.

References

1. Li H., Hestenes DF. and Rockwood A., *Generalized Homogeneous coordinates for computational geometry*. In G. Somer, editor, *Geometric Computing with Clifford Algebras*, pages 27-52. Springer-Verlag Heidelberg, 2001.
2. Bayro-Corrochano Eduardo, *Geometric Computing: for Wavelet Transforms, Robot Vision, Learning, Control and Action*. Springer Verlag, Jan, 2010.
3. J. Zamora and E. Bayro-Corrochano, *Kinematics and Differential Kinematics of Binocular Heads*. International Conference of Robotics and Automation (ICRA'06), Orlando, USA, 2006.
4. H. Khalil *Nonlinear Systems*. Prentice-Hall, Ch. 3-5, USA, 1996.
5. www.mathworks.com/products/matlab/
6. http://www.clucalc.info/ Mathematical tool to Geometric Algebra
7. V. I. Utkin, J. Guldner and J. Shi *Sliding Mode Control in Electromechanical Systems*. Taylor and Francis, 1999.
8. O. Carbajal, L. González, A. Loukianov and E. Bayro-Corrochano *Modeling and control of a humanoid arm using Conformal Geometric Algebra and sliding modes*. 2011 International Conference on Humanoid Robots, Bled, Slovenia, 2011, pp. 389 – 394.
9. L. González, A. Loukianov and E. Bayro-Corrochano *Integral Nested Super-Twisting Algorithm for Robotic Manipulators*. 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Taipei, Taiwan, 2010, pp. 3580 – 3585.
10. Jaime A. Moreno and Marisol Osorio, *A Lyapunov approach to second-order sliding mode controllers and observers*.Proceedings of the 47th IEEE Conference on Decision and Control, Cancún, México, 2008, pp. 2856 – 2856.
11. Mark W. Spong, Seth Hutchinson and M. Vidyasagar, *Robot Modeling and Control*. John Wiley and Sons, Inc., 2005.
12. J. Zamora and E. Bayro-Corrochano, *Parallel forward Dynamics: A geometric approach* . 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Taipei, Taiwan, 2010, pp. 2377 – 2382.