

Locomoción Humanoide Basada en Estrategias de Control Servo Visual

Josafat Delfín* Gustavo Arechavaleta* Héctor M. Becerra**

* *Robótica y Manufactura Avanzada, CINVESTAV, Unidad Saltillo, Av. Industria Metalúrgica 1064, Ramos Arizpe, Coah., México, C.P. 25903, (e-mail: {josafat.delfin,garechav}@cinvestav.edu.mx).*

** *Centro de Investigación en Matemáticas (CIMAT), Jalisco S/N, Col. Valenciana, C.P. 36240, Guanajuato, Gto., Mexico (e-mail: hector.becerra@cimat.mx).*

Resumen— Este trabajo aborda el problema de la generación de marcha humanoide con retroalimentación visual. En particular, la tarea que debe resolver el robot humanoide se define en términos de un error entre un par de imágenes, la actual y una objetivo. Se propone utilizar la cámara que se encuentra montada en la cabeza del robot para adaptar dos enfoques de control servo-visual que relacionen el error entre imágenes con la generación de la marcha humanoide. Por un lado, las estrategias de control visual que se adoptaron permiten desplazamientos omidireccionales de la cámara sin requerir información 3D del entorno. Por otro lado, el enfoque de generación de marcha calcula de forma reactiva el siguiente contacto de la pata en movimiento, sujeta a las restricciones de equilibrio dinámico, dada una velocidad de referencia para el centro de masa del robot. La evaluación experimental del acoplamiento de la marcha humanoide con retroalimentación visual se llevó a cabo con el robot humanoide NAO.

Keywords: Locomoción Humanoide, Control Visual

1. INTRODUCCIÓN

El problema de locomoción humanoide ha sido ampliamente estudiado en la literatura reciente. La complejidad radica principalmente en mantener el equilibrio dinámico durante la marcha mientras se genera la coordinación del movimiento del mecanismo. Comúnmente, los generadores de marcha humanoide (GMH) suponen referencias predefinidas del punto de momento nulo (PMN) o, en su defecto, la secuencia de pasos que debe reproducir el robot durante la coordinación de su movimiento (ver Kajita et al. (2003)). En Herdt et al. (2010) proponen un esquema basado en programación cuadrática (PC) que permite generar la secuencia de pasos de forma reactiva. En dicho enfoque, la entrada es una referencia en velocidad lineal del centro de masa (CdM) del humanoide. Las salidas son la trayectoria del CdM que cumple con el criterio del equilibrio dinámico y la trayectoria de la pata que debe desplazarse hacia el siguiente punto de contacto con el piso.

La tarea de locomoción asignada al robot puede definirse en términos de un objetivo visual. Esto implica que los espacios de control postural y pose del robot así como los espacios donde habitan las tareas, incluyendo el espacio de la imagen, deben interactuar. En Dune et al. (2010) proponen un esquema de control visual que considera el error de balanceo lateral inherente a la marcha bípeda dentro del esquema de locomoción sugerido en Herdt et al. (2010). En este caso, la marcha se genera a partir de un error visual en la imagen y la velocidad lineal resultante de la cámara la transforman en la referencia de velocidad del CdM que, a su vez, se convierte en la entrada para



Figura 1. Tarea de posicionamiento de un robot humanoide mediante locomoción basada en estrategias de control visual.

el GMH basado en PC. Todos los experimentos realizados en los trabajos antes mencionados se llevaron a cabo con la plataforma humanoide HRP-2 que posee una escala humana.

En este artículo se propone un enfoque similar con la plataforma humanoide NAO que posee una altura aproximada de 60 cm y porta un sistema visual monocular ubicado en su cabeza. La tarea que debe realizar el robot consiste en caminar desde una ubicación inicial hasta otra final utilizando únicamente información visual detectada sobre un plano observado durante el recorrido. Este escenario se presenta en la Figura 1, donde se han usado dos robots distintos para marcar la ubicación inicial y final. En los

experimentos realizados, el robot adquiere una imagen del plano observado desde la ubicación final. Después, se reubica espacialmente a una distancia determinada de forma manual. Desde la nueva ubicación también observa el mismo plano. En consecuencia, las dos imágenes adquiridas comparten información visual. A partir de esta información visual compartida, se construye un error visual que debe regularse hacia una vecindad cercana a cero mediante un controlador servo-visual. Las consignas que produce el control servo-visual conforman la entrada para el GMH. En conjunto, ambos logran resolver la tarea de alcanzar la ubicación deseada.

La contribución derivada de este trabajo es la evaluación de distintas estrategias de control visual en tareas de locomoción humanoide. En particular, se compara el desempeño de dos diferentes tipos de controladores visuales: un control basado en posición y otro basado en imagen. Los dos controladores visuales existentes en la literatura (Chaumette y Hutchinson (2006), y Benhimane y Malis (2007)) poseen la ventaja de no requerir información relacionada con la estructura de la escena observada, como pudiera ser profundidad o un modelo 3D del plano observado.

2. LOCOMOCIÓN HUMANOIDE

La generación de la locomoción humanoide involucra las siguientes etapas:

1. Planificar una trayectoria para un modelo de control reducido.
2. Planificar la secuencia de pasos.
3. Generar una referencia del PMN-CdM factible.
4. Generar la coordinación del movimiento postural.

En el presente trabajo se utiliza un GMH reactivo de tal modo que las etapas 1 y 2 no son necesarias. El problema de la etapa 3 se formula con el enfoque de control predictivo basado en modelo lineal. Esta formulación permite agregar restricciones lineales de desigualdad con el fin de representar el área factible para posicionar la pata respetando el criterio del PMN. La etapa 4 se resuelve con PC jerárquica para generar la coordinación del movimiento postural correspondiente a la marcha.

2.1 Modelo y generación de la marcha

En Kajita et al. (2003) el modelo del carro sobre una mesa se propuso para capturar la dinámica representativa de la marcha bípeda en términos de un sistema lineal del PMN que se puede deducir a partir de la secuencia de pasos. Esta secuencia puede generarse a partir de una trayectoria predefinida o puede calcularse con un planificador de pasos (Chestnutt et al., 2005). La entrada es entonces la referencia del PMN y la salida debe corresponder a la trayectoria del CdM. La versión simplificada del sistema dinámico discreto es de la forma:

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k), \\ \mathbf{p}(k) &= \mathbf{c}\mathbf{x}(k), \end{aligned} \quad (1)$$

y las variables que están involucradas corresponden a:

$$\begin{aligned} \mathbf{x}(k) &= [x(kT) \dot{x}(kT) \ddot{x}(kT)]^T, \\ \mathbf{u}(k) &= u_x(kT), \quad \mathbf{p}(k) = p_x(kT), \\ \mathbf{A} &= \begin{bmatrix} 1 & T & T^2/2 \\ 0 & 1 & T \\ 0 & 0 & T \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} T^3/6 \\ T^2/2 \\ T \end{bmatrix}, \quad \mathbf{c} = [1 \ 0 \ -h/g], \end{aligned}$$

donde T es el periodo de muestreo, el subíndice x representa la posición del CdM con respecto al eje x (desplazamiento hacia adelante) ya que el análisis con el eje y es idéntico. El *jerk* del CdM está representado por $u_x = \ddot{x}$ mientras que p_x describe la posición del PMN. La altura con respecto al piso y la norma de la fuerza de gravedad son h y g , respectivamente. Las ecuaciones (1) pueden considerarse durante un horizonte de tiempo determinado por los futuros N pasos. En este caso, la dinámica del sistema se puede expresar en el intervalo de tiempo NT para relacionar \ddot{x} con p_x :

$$\dot{\mathbf{x}}(k+1) = \begin{bmatrix} \dot{\mathbf{x}}(k+1) \\ \vdots \\ \dot{\mathbf{x}}(k+N) \end{bmatrix} = \bar{\mathbf{A}}\mathbf{x}(k) + \bar{\mathbf{B}}\mathbf{u}(k), \quad (2)$$

$$\mathbf{p}(k+1) = \begin{bmatrix} p(k+1) \\ \vdots \\ p(k+N) \end{bmatrix} = \mathbf{C}_x\mathbf{x}(k) + \mathbf{C}_u\mathbf{u}(k), \quad (3)$$

y $\mathbf{u}(k)$, $\bar{\mathbf{A}}$, $\bar{\mathbf{B}}$, \mathbf{C}_x y \mathbf{C}_u definidas de la siguiente forma:

$$\begin{aligned} \mathbf{u}(k) &= \begin{bmatrix} u(k) \\ \vdots \\ u(k+N-1) \end{bmatrix}, \\ \bar{\mathbf{A}} &= \begin{bmatrix} 0 & 1 & T \\ \vdots & \vdots & \vdots \\ 0 & 1 & NT \end{bmatrix}, \quad \bar{\mathbf{B}} = \begin{bmatrix} T^2/2 & 0 & 0 \\ \vdots & \ddots & 0 \\ (1+2N)T^2/2 & \dots & T^2/2 \end{bmatrix}, \\ \mathbf{C}_x &= \begin{bmatrix} 1 & T & T^2/2 - h/g \\ \vdots & \vdots & \vdots \\ 1 & NT & N^2T^2/2 - h/g \end{bmatrix}, \\ \mathbf{C}_u &= \begin{bmatrix} T^3/6 - Th/g & 0 & 0 \\ \vdots & \ddots & 0 \\ (1+3N+3N^2)T^3/6 - Th/g & \dots & T^3/6 - Th/g \end{bmatrix}. \end{aligned}$$

A partir de las ecuaciones (2) y (3) se puede formular un PC con horizonte de tiempo (Wieber, 2006). Este problema corresponde a minimizar el *jerk* del CdM junto con el error en posición del PMN \mathbf{p} y su referencia \mathbf{p}_r tal que:

$$\min_{\mathbf{u}(k)} \frac{\alpha}{2} \|\mathbf{u}(k)\|^2 + \frac{\gamma}{2} \|\mathbf{p}(k+1) - \mathbf{p}_r(k+1)\|^2. \quad (4)$$

Esta formulación supone una referencia del PMN previamente calculada con un planificador de pasos (Gouaillier et al., 2010). En Herdt et al. (2010) eliminan esta suposición mediante la regulación de la velocidad del CdM hacia un valor deseado $\dot{\mathbf{x}}_r$. Considerando la posición actual de la pata en contacto $\mathbf{x}_c(k)$, las posiciones de los siguientes pasos $\mathbf{x}_f(k)$ se pueden adaptar automáticamente utilizando matrices de selección $\mathbf{S}_c(k+1)$ y $\mathbf{S}(k+1)$ e incorporando restricciones lineales de desigualdad que delimiten el área factible para posicionar las patas sucesivamente. El PC en (4) se puede reformular de la siguiente manera:

$$\min_{\bar{\mathbf{u}}(k)} \frac{\alpha}{2} \|\bar{\mathbf{u}}(k)\|^2 + \frac{\beta}{2} \|\dot{\bar{\mathbf{x}}}(k+1) - \dot{\bar{\mathbf{x}}}_r(k+1)\|^2 + \frac{\gamma}{2} \|\mathbf{p}(k+1) - \mathbf{p}_r(k+1)\|^2, \quad (5)$$

sujeto a

$$\mathbf{D}(k+1)\bar{\mathbf{u}}(k) \leq \mathbf{b}(k+1)$$

donde

$$\bar{\mathbf{u}}(k) = \begin{bmatrix} \mathbf{u}(k) \\ \mathbf{x}_f(k) \end{bmatrix}, \mathbf{p}_r(k+1) = \mathbf{S}_c(k+1)\mathbf{x}_c(k) + \mathbf{S}(k+1)\mathbf{x}_f(k).$$

La matriz $\mathbf{D}(k+1)$ y el vector $\mathbf{b}(k+1)$ de coeficientes representan las restricciones lineales de desigualdad.

2.2 Generación del movimiento postural

El movimiento postural del humanoide se refiere a los desplazamientos coordinados de sus articulaciones. El vector de parámetros articulares (independientes) $\mathbf{q} \in \mathcal{CS}$ habita en el espacio de configuración \mathcal{CS} de dimensión $n = \dim(\mathcal{CS})$. Las posiciones del CdM y de la pata en movimiento están definidas en el espacio de la tarea \mathcal{OS} , donde $\mathbf{f} \in \mathcal{OS}$ es un punto sobre \mathcal{OS} y $m = \dim(\mathcal{OS}) \leq 6$ es su dimensión. Si \mathbf{q} representa las coordenadas articulares del mecanismo, entonces se puede definir un error de la tarea en términos de \mathbf{q} :

$$\mathbf{e}(\mathbf{q}) = (\mathbf{f}(\mathbf{q}) - \mathbf{f}^d) \rightarrow 0.$$

En este caso, $\mathbf{f}(\mathbf{q})$ representa la cinemática directa. La derivada con respecto al tiempo del error de la tarea cinemática es

$$\dot{\mathbf{e}}(\mathbf{q}) = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}}.$$

donde $\mathbf{J}(\mathbf{q}) = \frac{\partial \mathbf{f}}{\partial \mathbf{q}}$ se conoce como la Jacobiana de la tarea. Si se impone una convergencia exponencial del error resulta que:

$$\dot{\mathbf{e}}(\mathbf{q}) = -\lambda \mathbf{e}(\mathbf{q}),$$

donde λ es un escalar positivo. Entonces

$$\mathbf{J}(\mathbf{q})\dot{\mathbf{q}} = -\lambda \mathbf{e}(\mathbf{q}) \rightarrow 0,$$

Como el robot humanoide NAO tiene más grados de libertad (articulaciones) n que restricciones de la tarea m , entonces $\mathbf{q} \in \mathbb{R}^n$ y $\mathbf{e}(\mathbf{q}) \in \mathbb{R}^m$ donde $m < n$. En este caso, la generación del movimiento cinemático equivale a minimizar un PC de la forma:

$$\min_{\dot{\mathbf{q}}} \frac{1}{2} \dot{\mathbf{q}}^T \dot{\mathbf{q}},$$

$$\text{s.t. } \mathbf{J}\dot{\mathbf{q}} - \dot{\mathbf{e}} = 0.$$

La solución que minimiza la norma de $\dot{\mathbf{q}}$ es

$$\dot{\mathbf{q}}^* = \mathbf{J}^+ \dot{\mathbf{e}},$$

donde $\mathbf{J}^+ = \mathbf{J}^T(\mathbf{J}\mathbf{J}^T)^{-1}$ es la pseudo-inversa de Moore-Penrose. Sin embargo, el conjunto solución completo es:

$$\dot{\mathbf{q}}^* = \mathbf{J}^+ \dot{\mathbf{e}}_1 + \mathbf{Q}\mathbf{z},$$

donde $\mathbf{Q} = \mathbf{I} - \mathbf{J}^+ \mathbf{J}$ es el proyector al espacio nulo de \mathbf{J} y \mathbf{z} es cualquier vector en \mathbb{R}^n . Esto permite resolver una tarea secundaria en el espacio nulo:

$$\min_{\dot{\mathbf{q}}_2} \frac{1}{2} \dot{\mathbf{q}}_2^T \dot{\mathbf{q}}_2,$$

$$\text{s.t. } \mathbf{J}_2 \mathbf{Q}_1 \dot{\mathbf{q}}_2 - (\dot{\mathbf{e}}_2 - \mathbf{J}_2 \mathbf{J}_1^+ \dot{\mathbf{e}}_1) = 0.$$

En este caso, la solución del segundo problema es:

$$\dot{\mathbf{q}} = \dot{\mathbf{q}}_1^* + (\mathbf{J}_2 \mathbf{Q}_1)^+ (\dot{\mathbf{e}}_2 - \mathbf{J}_2 \dot{\mathbf{q}}_1^*).$$

Este procedimiento se puede extender fácilmente para resolver simultáneamente p tareas con un robot humanoide (Kanoun et al., 2011). El siguiente algoritmo genera el movimiento postural considerando varias tareas cinemáticas con jerarquías: posición y orientación de la pata en contacto, posición del CdM, posición y orientación de la pata en movimiento, evasión de límites articulares y autocolisiones, etc.

Algoritmo 1

Entrada: $\mathbf{f}_{\{1..p\}}^d, \mathbf{q}_0, \Delta t$.

Salida: la ruta articular \mathbf{q} tal que $\|\mathbf{e}(\mathbf{q})\| < TOL$.

1. $j \leftarrow 1$
2. Calcular $\mathbf{f}(\mathbf{q}_j)$ y $\mathbf{e}(\mathbf{q}_j)$
3. **Mientras** $\|\mathbf{e}_{\{1..p\}}(\mathbf{q}_j)\| > TOL$ ó $j \leq \text{maxTry}$
4. Calcular $\mathbf{J}_i(\mathbf{q}_j)$ para $i = 1 \dots p$
5. $\dot{\mathbf{q}}_j^* \leftarrow \sum_{i=1}^p \bar{\mathbf{J}}_i^+ (\dot{\mathbf{e}}_i - \mathbf{J}_i \dot{\mathbf{q}}_{i-1}^*) + \mathbf{Q}_p \mathbf{z}$
6. $\mathbf{q}_{j+1} \leftarrow \mathbf{q}_j + \dot{\mathbf{q}}_j^* \Delta t$
7. Calcular $\mathbf{f}(\mathbf{q}_{j+1})$ y $\mathbf{e}(\mathbf{q}_{j+1})$
8. $j \leftarrow j + 1$
9. **fin**

donde $\bar{\mathbf{J}}_i = \mathbf{J}_i \mathbf{Q}_{i-1}$, $\mathbf{Q}_i = \mathbf{Q}_{i-1} - \bar{\mathbf{J}}_i^+ \bar{\mathbf{J}}_i$, $\mathbf{Q}_0 = \mathbf{I}_n$ y TOL representa la tolerancia mínima del error.

3. FUNDAMENTOS DE CONTROL VISUAL

El control servo-visual se refiere al uso de la información generada mediante visión por computadora para controlar el movimiento del robot (Chaumette y Hutchinson (2006)). El objetivo de todos los esquemas de control basados en visión es minimizar un error $\mathbf{e}(t)$ definido como

$$\mathbf{e}(t) = \mathbf{s}(t) - \mathbf{s}^d \quad (6)$$

donde $\mathbf{s}(t)$ es un vector de k medidas de imagen observadas desde la ubicación actual de la cámara, por lo que tienen una dependencia temporal. El vector \mathbf{s}^d contiene los valores deseados de las medidas de imagen, esto es, la referencia visual, que en este trabajo se considerará constante. Los esquemas de control servo-visual difieren principalmente en la manera en que \mathbf{s} se expresa. Comúnmente existen dos enfoques generales, el primero se conoce como control servo-visual basado en imagen (IBVS, por sus siglas en inglés), donde \mathbf{s} representa a un conjunto de medidas de imagen definidas directamente en píxeles. El segundo es el control servo-visual basado en posición (PBVS, por sus siglas en inglés), donde \mathbf{s} es un conjunto de parámetros 3D (posición y orientación), los cuales deben ser estimados a partir de la información en las imágenes.

Un control en velocidad, por ejemplo, consiste en encontrar la relación entre la variación en el tiempo de \mathbf{s} con respecto a la velocidad de la cámara $\boldsymbol{\nu}_c = (\mathbf{v}_c, \boldsymbol{\omega}_c)$, donde $\mathbf{v}_c = (v_x, v_y, v_z)^T$ es la velocidad lineal y $\boldsymbol{\omega}_c = (\omega_x, \omega_y, \omega_z)^T$ la velocidad angular. Esta relación se puede escribir como:

$$\dot{\mathbf{s}} = \mathbf{L}_s \boldsymbol{\nu}_c,$$

donde $\mathbf{L}_s \in \mathbb{R}^{k \times 6}$ se conoce como la matriz de interacción.

4. ESQUEMA DE CONTROL VISUAL HUMANOIDE

El esquema de locomoción necesita una trayectoria de referencia del CdM para calcular automáticamente las tareas

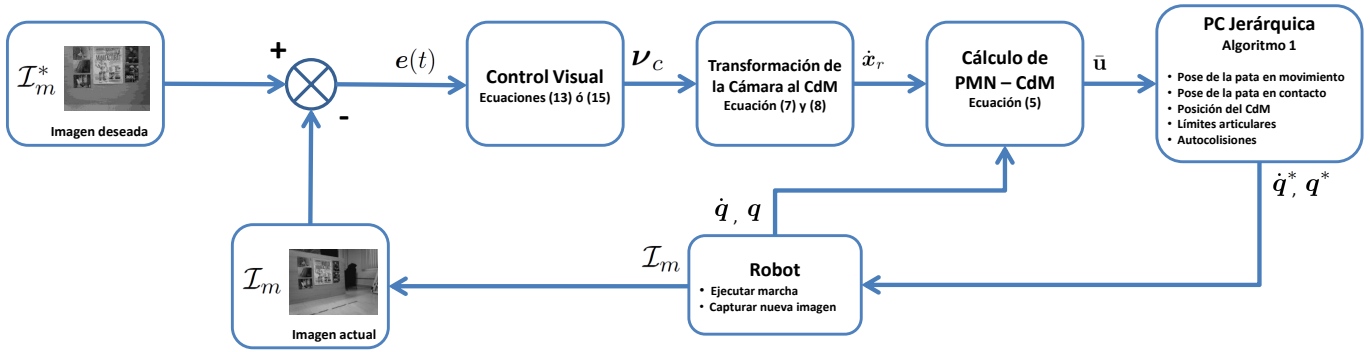


Figura 2. Esquema de Control Visual Humanoide.

cinemáticas que resuelven el movimiento de la marcha considerando las restricciones del PMN. En consecuencia, la velocidad de la cámara calculada mediante un esquema de control visual puede ser transformada para que corresponda a la velocidad de referencia del CdM que requiere el GMH (ver Figura 2).

Sea ν_c la velocidad del sistema de referencia cámara \mathcal{C} y ν_m la velocidad del sistema de referencia asociado al CdM del robot \mathcal{M} . Suponiendo que entre los sistemas de referencia \mathcal{C} y \mathcal{M} existe una transformación espacial constante mT_c , entonces se cumple la siguiente relación entre las velocidades:

$$\nu_m = {}^mT_c \nu_c, \quad (7)$$

donde

$${}^mT_c = \begin{pmatrix} {}^mR_c^T & -{}^mR_c^T[r]_\times \\ 0 & {}^mR_c^T \end{pmatrix} \in \mathbb{R}^{6 \times 6},$$

donde $[r]_\times \in \mathbb{R}^{3 \times 3}$ representa una matriz antisimétrica y ${}^mR_c \in SO(3)$. Esta relación es válida considerando que la cabeza está fija para tener una vista frontal durante la locomoción. Una vez que la velocidad de cámara se expresa en el sistema de referencia del CdM, se construye la entrada del GMH:

$$\dot{x}_r = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \nu_m \quad (8)$$

que interviene en el problema (5). Cabe resaltar que el GMH sólo considera tres de las seis componentes de velocidad contenidas en ν_m , a saber: v_x, v_y y ω_z . Estas son las velocidades lineal longitudinal, lineal lateral y angular respecto a la vertical.

En las siguientes secciones, se describen dos esquemas de control visual, uno basado en posición y otro basado en imagen, que permiten calcular la velocidad de referencia ν_m . Ambos controles utilizan una restricción geométrica estimada a partir de medidas de imagen correspondientes. Esta restricción se introduce brevemente a continuación.

Sean p_i y p_i^* las coordenadas homogéneas expresadas en píxeles para un punto en las imágenes actual \mathcal{I}_m y deseada \mathcal{I}_m^* , que corresponden a la proyección de un punto 3D denotado por \mathcal{P} . Los puntos de imagen normalizado están dados como $m_i = K^{-1}p_i$ y $m_i^* = K^{-1}p_i^*$ (ver Figura 3). Entonces, p_i está relacionado con p_i^* mediante $p_i = G_i p_i^*$, donde G_i es una matriz homografía. Si todos los puntos de interés caen en un plano 3D denotado por π , entonces hay una única matriz homografía G tal que

$$p_i = G p_i^*, \quad (9)$$

para todo i . La matriz G es una homografía proyectiva y se puede estimar a partir de cuatro puntos correspondientes en las imágenes actual \mathcal{I}_m y deseada \mathcal{I}_m^* . Si no todos los puntos pertenecen al mismo plano 3D, entonces tres puntos pueden usarse para definir un plano y cinco puntos adicionales son necesarios para estimar G (Malis et al. (2000)).

Al remover el efecto proyectivo dada la matriz de calibración K , se obtiene una matriz homografía Euclidiana:

$$H = K^{-1} G K, \quad (10)$$

la cual cumple con la restricción $m_i = H m_i^*$. La matriz homografía Euclidiana codifica la transformación rígida existente entre los sistemas de referencia asociados a la cámara actual \mathcal{C} y deseada \mathcal{C}^* como sigue:

$$H = R + \frac{t}{d^*} n^{*T}, \quad (11)$$

donde R es la matriz de rotación que alinea el sistema de referencia actual \mathcal{C} con el sistema de referencia deseado \mathcal{C}^* , el vector t es la traslación entre sistemas de referencia expresado en \mathcal{C}^* , n^* es el vector unitario normal al plano π expresado en \mathcal{C}^* y d^* es la distancia desde π a \mathcal{C}^* . Ver la Figura 3 para mayor detalle.

4.1 Control basado en posición (control 3D)

Sea t el vector de traslación entre los sistemas de referencia actual \mathcal{C} y deseado \mathcal{C}^* expresado en \mathcal{C}^* , y R la matriz de rotación entre ellos. Se plantea un control basado en posición utilizando un vector de medidas actual:

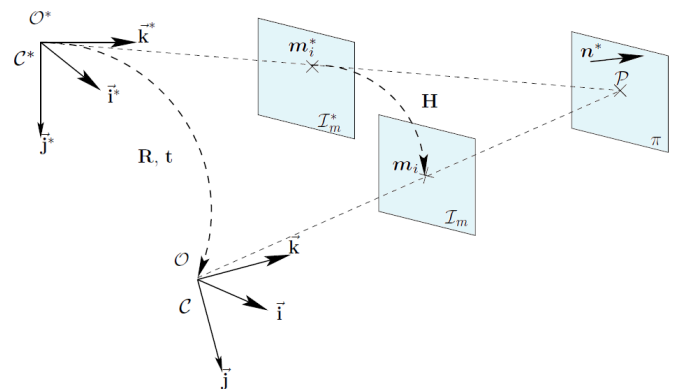


Figura 3. Modelo de proyección y homografía entre dos imágenes de un plano.

$$\mathbf{s} = (\mathbf{t}, \theta \hat{\mathbf{u}}) \in \mathbb{R}^6, \quad (12)$$

donde $\theta \hat{\mathbf{u}}$ representa la parametrización eje/ángulo de la matriz de rotación (formulada de Rodrigues). Dado que este vector de medidas \mathbf{s} expresa la traslación y rotación relativas al sistema de referencia deseado \mathcal{C}^* , en este caso tenemos $\mathbf{s}^d = 0$ y consecuentemente $\mathbf{e} = \mathbf{s}$. La derivada temporal del vector de error resulta en $\dot{\mathbf{e}} = \mathbf{L}_s \mathbf{v}_c$, donde la matriz de interacción es una matriz cuadrada.

Las velocidades de la cámara resultan desacopladas para este planteamiento y están dadas como sigue:

$$\begin{aligned} \mathbf{v}_c &= -\lambda \mathbf{R}^T \mathbf{t}, \\ \boldsymbol{\omega}_c &= -\lambda \theta \hat{\mathbf{u}}. \end{aligned} \quad (13)$$

Considerando la ganancia del control $\lambda > 0$, se consigue una dinámica exponencialmente estable $\dot{\mathbf{e}} = -\lambda \mathbf{e}$.

Es posible descomponer \mathbf{H} según (11) para obtener \mathbf{R} , \mathbf{t} y \mathbf{n} , los cuales son necesarios para implementar la ley de control (13). Un algoritmo eficiente que lleva a cabo la descomposición de \mathbf{H} se puede consultar en Triggs (1998). Como es bien conocido, este tipo de descomposiciones generan dos soluciones geoméricamente válidas, sin embargo, sólo una de ellas es físicamente admisible. La solución correcta se puede seleccionar tomando la solución asociada al vector normal cuya tercera componente (n_z) sea mayor.

4.2 Control basado en imagen (control 2D)

En esta sección se describe un control basado en imagen que directamente utiliza la matriz de homografía para definir un vector de errores (función de tarea) de traslación y rotación que se introducen en la ley de control. La idea central en este esquema, desarrollado en Benhimane y Malis (2007), es que los dos sistemas de referencia \mathcal{C} y \mathcal{C}^* coinciden, si y sólo si, la matriz homografía \mathbf{H} es igual a la matriz identidad \mathbf{I} . Siguiendo esta idea, se construye una función de tarea $\mathbf{e} \in \mathbb{R}^6$ localmente isomórfica a la pose (traslación y rotación) de la cámara. La función de tarea $\mathbf{e} = (\mathbf{e}_v^T, \mathbf{e}_\omega^T)^T$ es nula, si y sólo si la cámara alcanza su pose de referencia y está definida como sigue:

$$\begin{aligned} \mathbf{e}_v &= (\mathbf{H} - \mathbf{I}) \mathbf{m}^*, \\ [\mathbf{e}_\omega]_\times &= \mathbf{H} - \mathbf{H}^T, \end{aligned} \quad (14)$$

donde \mathbf{m}^* es cualquier punto en la imagen objetivo \mathcal{I}_m^* que pertenece al plano que define \mathbf{H} , y $[\mathbf{e}_\omega]_\times$ representa la matriz antisimétrica obtenida a partir del vector $\mathbf{e}_\omega = (e_{\omega x}, e_{\omega y}, e_{\omega z})^T$.

La función de tarea \mathbf{e} puede ser estimada usando solamente las dos imágenes \mathcal{I}_m y \mathcal{I}_m^* sin necesidad de obtener la estructura 3D de la ubicación objetivo (\mathbf{n}^* y d^*). Sólo es necesario estimar \mathbf{H} a partir de las ecuaciones (9) y (10). La derivada temporal de la función de tarea genera la ecuación $\dot{\mathbf{e}} = \mathbf{L} \mathbf{v}_c$, donde \mathbf{L} es una matriz de interacción cuadrada. En Benhimane y Malis (2007) se demuestra que no es necesario para este esquema de control hacer uso de la matriz de interacción para definir la ley de control. El control lineal

$$\begin{aligned} \mathbf{v}_c &= -\lambda \mathbf{e}_v, \\ \boldsymbol{\omega}_c &= -\lambda \mathbf{e}_\omega, \end{aligned} \quad (15)$$

con $\lambda > 0$, consigue que el error tenga una dinámica localmente estable. Esta ley de control sólo depende de la función de tarea, la cual converge exponencialmente a $\mathbf{0}$. La estabilidad local de la dinámica de error se garantiza para cualquier estructura 3D y para cualquier punto \mathbf{m}^* .

5. EXPERIMENTOS

La validación experimental de los controles presentados se realizó en un robot humanoide NAO. En los experimentos se utilizó la cámara superior montada sobre la cabeza del robot, donde además no existe ninguna rotación entre la cabeza y el torso. Las imágenes usadas para la implementación de los controles son obtenidas a partir de un video con una frecuencia de muestreo de 12 Hz y una resolución de 640×480 . Para obtener la información necesaria en el control servo-visual se utilizó un algoritmo de detección y descripción de puntos, búsqueda de correspondencias entre la imagen objetivo y la imagen inicial y un algoritmo de seguimiento para medir cada punto en cada iteración. Para el proceso de detección se utilizó un detector de esquinas basado en Shi y Tomasi (1994) e implementado en la función `goodFeaturesToTrack` de la biblioteca OpenCV (2014). A cada punto se le asignó un descriptor de tipo SIFT (Lowe (1999)). Un emparejador robusto que utiliza el algoritmo RANSAC de Fischler y Bolles (1981) se aplicó para la búsqueda de las correspondencias. Finalmente se utilizó un algoritmo de seguimiento de puntos basado en el método iterativo de flujo óptico en pirámides Lucas-Kanade (Bouguet (2001)) que está implementado en la función `calcOpticalFlowPyrLK` de OpenCV. Los parámetros de trabajo del seguidor fueron ajustados experimentalmente para el tipo de marcha desempeñada por el robot NAO, es decir, debido al movimiento de balanceo de la cámara causados por cada paso, es necesario un ajuste adecuado de los parámetros del seguidor. Cabe destacar que el seguidor fue evaluado experimentalmente y se concluyó que es capaz de funcionar a la máxima velocidad de avance frontal, lateral y de rotación cuando el tamaño de paso máximo para cada dirección es 0.04m frontal, 0.14m lateral y 0.349rad en rotación (Aldebaran-Robotics (2014)).

Con relación a la evaluación de los controles, el robot fue calibrado con respecto a una cámara aérea para proveer una medida real de la tarea de posicionamiento (ver Figura 8). Primero, la imagen objetivo es capturada en la ubicación deseada. Como se describió en la introducción, la tarea de posicionamiento es ejecutada con respecto a un objeto planar, comenzando desde una posición inicial a partir de la cual es posible ver el objeto planar desde un ángulo diferente. El robot es controlado para alcanzar la ubicación deseada utilizando las leyes de control de las ecuaciones (13) y (15) definiendo $\lambda = 0,1$. La pose inicial se encuentra trasladada 1m y rotada 36° con respecto a la pose deseada. En las Figuras 4 - 7 se puede observar la transformación proyectiva de los puntos en la pose deseada (verdes) y los puntos actuales (rojos). En todos los experimentos se estableció la misma condición de paro relativa a la norma de los errores en las ecuaciones (13) y (15).

En lo que respecta a la calibración de la cámara, se utilizaron dos conjuntos de parámetros:

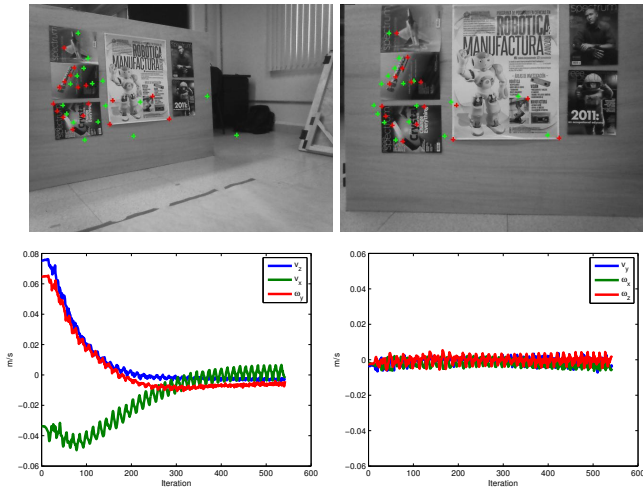


Figura 4. Control 3D con K . Primer renglón: imagen inicial e imagen final. Segundo renglón: velocidades de entrada al GMH y velocidades no requeridas por el GMH.

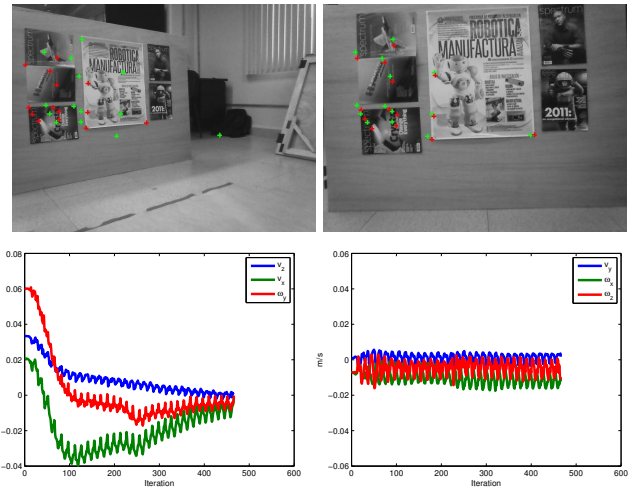


Figura 6. Control 2D con K . Primer renglón: imagen inicial e imagen final. Segundo renglón: velocidades de entrada al GMH y velocidades no requeridas por el GMH.

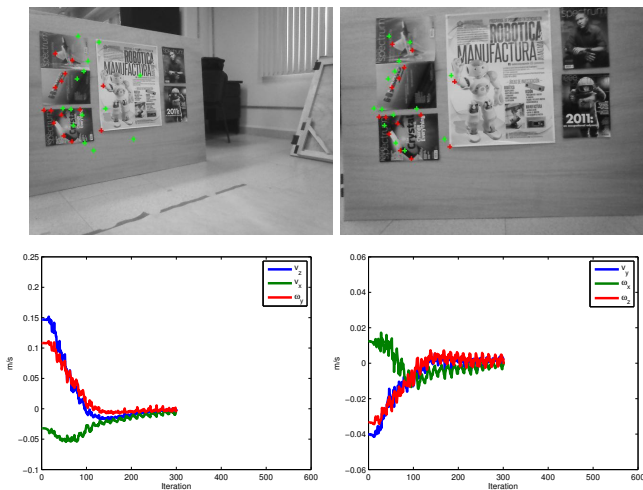


Figura 5. Control 3D con \hat{K} . Primer renglón: imagen inicial e imagen final. Segundo renglón: velocidades de entrada al GMH y velocidades no requeridas por el GMH.

- Buena calibración K : los parámetros utilizados fueron, $\alpha_x = 531.712$, $\alpha_y = 532.662$, $u_0 = 318.244$ y $v_0 = 250.698$; los cuales fueron obtenidos por las funciones de OpenCV.
- Mala calibración \hat{K} : los parámetros anteriores fueron alterados manualmente quedando como $\alpha_x = 712.4941$, $\alpha_y = 713.7671$, $u_0 = 600.4869$ y $v_0 = 470.9912$.

Cuadro 1. Errores finales en pose e imagen

Control	Traslación (m)	Rotación (grados)	Pixeles
3D con K	0.01	2.8	17.2
3D con \hat{K}	0.02	4.3	18.7
2D con K	0.04	2.5	11.6
2D con \hat{K}	0.03	1.7	10.4

Los resultados del control 3D para K y \hat{K} se muestran en las Figuras 4 y 5. El vector de velocidades ν_m se muestra en las Figuras 4 y 5, separando las velocidades que son entrada del GMH (izquierda) de las que no

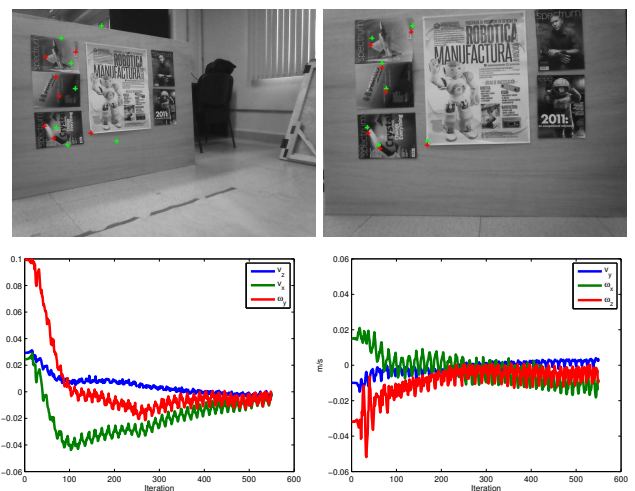


Figura 7. Control 2D con \hat{K} . Primer renglón: imagen inicial e imagen final. Segundo renglón: velocidades de entrada al GMH y velocidades no requeridas por el GMH.

son utilizadas (derecha). Lo anterior debido a que la locomoción sólo requiere las velocidades proyectadas en el plano de movimiento. En el caso de \hat{K} todas las velocidades resultan afectadas, aún las que no se utilizan en la locomoción. Sin embargo, aún con las perturbaciones generadas por la mala calibración, el control 3D termina convergiendo a una vecindad cercana a la pose deseada (ver Figura 8).

Los resultados para los controles 2D se muestran en las Figuras 6 y 7. Las imágenes están organizadas de la misma manera que para el control 3D. Tal como se menciona en Benhimane y Malis (2007), el control 2D es robusto a malas calibraciones permitiendo que el robot alcance una vecindad cercana a la pose deseada. En comparación con el control 3D, el control 2D resulta más robusto a malas calibraciones. Lo anterior puede observarse en la Figura 8, donde la trayectoria del robot usando \hat{K} en el control 3D resulta muy diferente a la trayectoria usando K , mientras

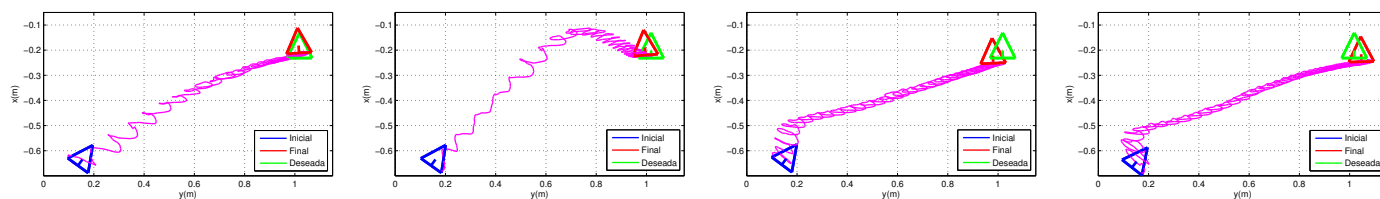


Figura 8. Tarea de posicionamiento del robot. Primera y segunda columna: control 3D utilizando K y \hat{K} . Tercera y cuarta columna: control 2D utilizando K y \hat{K} .

que el control 2D genera una trayectoria muy similar para ambos casos, con buena y mala calibración.

En el cuadro 1 se muestran los errores finales de posición, orientación y el error promedio final en píxeles de la proyección de los puntos deseados con los actuales. Podemos ver que aún con el constante derrapamiento de los pies del robot en el piso, en todos los casos el robot finaliza en una vecindad no mayor a 0.05m y con un error en orientación de menos de 5 grados. Estos errores son suficientes para aplicaciones que impliquen recorrer distancias mayores, por ejemplo navegación.

6. CONCLUSIONES

En este trabajo se mostró el desempeño de dos estrategias de control visual representativas en tareas de locomoción humanoide. Estos controles visuales se lograron acoplar con el generador de marcha humanoide (GMH) basado en programación cuadrática (PC). Esto fue posible gracias a que el GMH no necesita una secuencia de pasos predefinida, ni una referencia previamente calculada del punto de momento nulo (PMN). Como se puede verificar experimentalmente, lo único que se requiere como entrada es la referencia en velocidad del CdM del robot para que automáticamente se genere la siguiente huella de la pata que respete el desplazamiento del PMN y la coordinación del movimiento postural de la marcha. Aplicando una transformación entre marcos de referencia se puede obtener la referencia en velocidad del CdM, a partir de la velocidad de la cámara calculada por medio de un error visual.

Actualmente se está trabajando en la extensión de estos enfoques para resolver tareas de navegación visual donde una ruta está compuesta por varias imágenes de referencia.

REFERENCIAS

- Aldebaran-Robotics (2014). *NAO Documentation*. Aldebaran Robotics. URL <https://community.aldebaran-robotics.com/doc/>.
- Benhimane, S. y Malis, E. (2007). Homography-based 2D Visual Tracking and Servoing. *Int. Journal of Robotics Research*, 26(7), 661–676.
- Bouguet, J.Y. (2001). Pyramidal implementation of the affine Lucas Kanade feature tracker description of the algorithm. *Intel Corporation*, 2, 3.
- Chaumette, F. y Hutchinson, S. (2006). Visual Servo Control I Basic Approaches. *Robotics and Automation Magazine*, 13(6), 82–90.
- Chestnutt, J., Lau, M., Cheung, G., Kuffner, J., Hodgins, J., y Kanade, T. (2005). Footstep Planning for the Hon- da Asimo Humanoid. In *IEEE International Conference on Robotics and Automation*, 629–634.
- Dune, C., Herdt, A., Stasse, O., Wieber, P.B., Yokoi, K., y Yoshida, E. (2010). Cancelling the Sway Motion of Dynamic Walking in Visual Servoing. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 3175–3180.
- Fischler, M.A. y Bolles, R.C. (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6), 381–395.
- Gouaillier, D., Collette, C., y Kilner, C. (2010). Omni-directional Closed-loop Walk for NAO. In *IEEE-RAS International Conference on Humanoid Robots*, 448–454.
- Herdt, A., Holger, D., Wieber, P.B., Dimitrov, D., Mombaur, K., y Moritz, D. (2010). Online walking motion generation with automatic foot step placement. *Advanced Robotics*, 24(5-6).
- Kajita, S., Kanehiro, F., Fujiwara, K., Harada, K., Yokoi, K., y Hirukawa, H. (2003). Biped Walking Pattern Generation by Using Preview Control of Zero-Moment Point. In *IEEE International Conference on Robotics and Automation*, 1620–1626.
- Kanoun, O., Lamiraux, F., y Wieber, P.B. (2011). Kinematic Control of Redundant Manipulators: Generalizing the task-priority framework to inequality task. *IEEE Transactions on Robotics*, 27(4), 785–792.
- Lowe, D.G. (1999). Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, volume 2, 1150–1157.
- Malis, E., Chaumette, F., y Boudet, S. (2000). 2 1/2 Visual Servoing with Respect to Unknown Objects through a New Estimation Scheme of Camera Displacement. *Int. Journal of Computer Vision*, 37(1), 79–97.
- OpenCV (2014). *OpenCV Documentation*. Willow Garage. URL <http://docs.opencv.org/>.
- Shi, J. y Tomasi, C. (1994). Good features to track. In *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR '94., 1994 IEEE Computer Society Conference on*, 593–600.
- Triggs, B. (1998). Autocalibration from Planar Scenes. In H. Burkhardt y B. Neumann (eds.), *Computer Vision - ECCV'98*, volume 1406 of *Lecture Notes in Computer Science*, 89–105. Springer Berlin Heidelberg.
- Wieber, P. (2006). Trajectory free linear model predictive control for stable walking in the presence of strong perturbation. In *IEEE International Conference on Humanoids*, 137–142.