



**UNIVERSIDAD DE CHILE  
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS  
DEPARTAMENTO DE INGENIERÍA MECÁNICA**

# **NUEVO MÉTODO DE AUTO-MODELADO FLEXIBLE PARA ROBOTS**

**MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO  
CIVIL MECÁNICO.**

**ISMAEL FERNÁNDEZ IZQUIERDO**

2011



**UNIVERSIDAD DE CHILE  
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS  
DEPARTAMENTO DE INGENIERÍA MECÁNICA**

# **NUEVO MÉTODO DE AUTO-MODELADO FLEXIBLE PARA ROBOTS**

**MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO  
CIVIL MECÁNICO.**

**ISMAEL FERNÁNDEZ IZQUIERDO**

**PROFESOR GUÍA:**

JUAN CRISTÓBAL ZAGAL MONTEALEGRE

**MIEMBROS DE LA COMISIÓN:**

JOSÉ DELPIANO COSTABAL  
WILLIAMS CALDERÓN MUÑOZ

SANTIAGO DE CHILE  
ABRIL 2011

RESUMEN DE LA MEMORIA  
PARA OPTAR AL TÍTULO DE  
INGENIERO CIVIL MECÁNICO  
POR: ISMAEL FERNÁNDEZ I.  
FECHA: 25/04/2011  
PROF. GUIA: JUAN CRISTÓBAL ZAGAL

## “NUEVO MÉTODO DE AUTO-MODELADO FLEXIBLE PARA ROBOTS”

En la actualidad, los sistemas de ingeniería tienen limitadas capacidades para adaptarse a una falla inesperada, y estas destrezas en general dependen de la habilidad del diseñador para prever dichas fallas. Sin embargo en los últimos años se han desarrollado algoritmos capaces de proveer a un robot de la capacidad de adaptarse a una falla inesperada (Bongard, Zykov y Lipson, 2006), (Zagaly Ruiz-del-Solar, 2004), (Bongard y Lipson, 2004). Estos algoritmos se basan en el concepto de auto-modelado continuo, donde se actualiza el modelo interno que tiene un robot de sí mismo y el modelo de funcionamiento (controlador) que utiliza para lograr un objetivo.

Este aprendizaje se logra gracias a una retroalimentación continua entre los datos que el robot recopila de sus sensores y un entorno de simulación física que selecciona modelos del robot que se ajusten al comportamiento registrado. Para el proceso de búsqueda de auto-modelos, es muy importante la utilización de una métrica para la comparación de señales entregadas por los sensores del robot. Esta métrica debe ser capaz de cuantificar cuan parecido es el comportamiento de un modelo candidato al del robot real.

En trabajos anteriores, como en (Bongard, Zykov y Lipson, 2006) y (Bongard y Lipson, 2004), se han implementado con éxito algoritmos de auto-modelado utilizando las métricas Euclidiana y Rolling Mean Metric. Sin embargo estas métricas tienen algunas limitaciones en cuanto a robustez. En particular son muy sensibles a pequeños cambios en la posición inicial de un robot, así como a pequeños desfases entre las señales a comparar.

En este trabajo se realiza un estudio comparativo de las métricas existentes y se presenta una nueva métrica llamada Discriminante de Kullback, de naturaleza estadística, que muestra considerables mejoras en cuanto a la robustez en la comparación de señales. También se expone la implementación de un algoritmo de auto-modelado incorporando la nueva métrica, obteniéndose buenos resultados en cuanto a la convergencia comparado con los resultados obtenidos con las métricas existentes.

Otra limitación de los trabajos realizados hasta el momento, como en (Bongard, Zykov y Lipson, 2006), (Zagaly Ruiz-del-Solar, 2004), (Bongard y Lipson, 2004), tiene que ver con los espacios de búsqueda utilizados. En particular en cuanto a las características geométricas de partes individuales de un robot, donde solo se ha utilizado formas simples como: Cubos, cilindros, esferas o capsulas.

En este trabajo se presenta una nueva representación genérica de robots, basada en un sólido amorfo definido por 6 puntos de control. Estos puntos son aproximados a través de una superficie de interpolación con splines cúbicos. Esta nueva representación amplía considerablemente el espacio de búsqueda para la geometría de partes individuales de un robot.

Se expone en este trabajo, la incorporación de esta nueva representación en un algoritmo de auto-modelado, obteniéndose buenos resultados en la convergencia y buena precisión a la hora de encontrar los parámetros del auto-modelo de un robot.

# Índice

1	Introducción .....	1
1.1	Antecedentes generales y motivación.....	1
1.2	Objetivos generales.....	3
1.3	Objetivos Específicos .....	3
2	Antecedentes .....	4
2.1	Auto-Modelado .....	4
2.1.1	Descripción.....	4
2.1.2	Etapas del auto-modelado .....	4
2.1.3	Síntesis de Movimiento.....	8
2.1.4	Métricas de comparación.....	9
2.1.5	Métrica Euclidiana .....	9
2.1.6	Métrica de Promedio Rodante (Rolling Mean Metric).....	10
2.2	Representación.....	12
2.2.1	Topología.....	12
2.2.2	Geometría.....	14
2.3	Representación a través de splines cúbicos.....	15
2.3.1	Splines .....	15
2.3.2	Interpolación de Hermite .....	15
2.3.3	Splines cardinales.....	16
2.4	Algoritmos genéticos .....	17
2.4.1	Individuos y Representación .....	17
2.4.2	Población.....	17
2.4.3	Selección.....	18
2.4.4	Cruce o Crossover .....	18
2.4.5	Mutación .....	18
2.5	Discriminante de Kullback.....	19
2.5.1	Definición para distribuciones continuas.....	19
2.5.2	Definición para distribuciones discretas (histogramas).....	19
2.5.3	Propiedades.....	19
2.6	Motor de Simulación física.....	20
2.6.1	Open Dynamics Engine (ODE).....	20
3	Metodología General de Trabajo .....	21

4	Resultados.....	22
4.1	Modelo de Simulación .....	22
4.1.1	Parámetros de la simulación: .....	22
4.1.2	Modelo robot trepen: .....	22
4.1.3	Dimensiones .....	22
4.1.4	Topología.....	23
4.1.5	Validación del modelo de simulación .....	24
4.2	Estudio de Métricas .....	25
4.2.1	Detalle experimento N° 1 .....	25
4.2.2	Series de tiempo .....	27
4.2.3	Métrica euclidiana .....	27
4.2.4	Métrica de promedio rodante (Rolling Mean Metric) .....	28
4.2.5	Nueva métrica propuesta.....	30
4.2.6	Discriminante de Kullback .....	30
4.2.7	Detalle experimento N° 2 .....	32
4.2.8	Detalle experimento N° 3.....	34
4.3	Algoritmo de auto-modelado incorporando la nueva métrica .....	38
4.3.1	Detalle experimento N° 3 .....	38
4.3.2	Detalle experimento N° 4.....	44
4.4	Nueva representación genérica de robots.....	47
4.4.1	Algunas representaciones posibles.....	49
4.4.2	Robot que incorpora la nueva representación.....	50
4.5	Algoritmo de auto modelado incorporando nueva representación .....	51
4.5.1	Detalle experimento N° 5 .....	51
4.5.2	Detalle experimento N° 6.....	54
5	Conclusiones .....	57
6	Bibliografía .....	58

# 1 Introducción

## 1.1 Antecedentes generales y motivación.

La habilidad de un robot de adaptar su comportamiento luego de sufrir una falla inesperada, podría tener importantes aplicaciones en la industria. Por ejemplo en robots semi-autónomos utilizados en cadenas de producción, robots controlados a distancia (pero que deben reaccionar rápidamente a cambios en su entorno), robots autónomos como los usados en la industria aeroespacial, satélites, sondas, robots exploradores, etc.

Sin embargo en la actualidad, los sistemas de ingeniería tienen limitada capacidad para adaptarse a fallas inesperadas, y estas destrezas en general dependen de la habilidad del diseñador para prever dichas fallas.

Gracias al desarrollo de algoritmos de auto-modelado, en los últimos años podemos aproximarnos al objetivo de crear robots más adaptables.

La idea es que un robot pueda desarrollar un modelo de sí mismo, utilizando solo datos extraídos de sus sensores durante el desarrollo de una tarea. Una vez que el robot obtiene un modelo de sí mismo (auto-modelo), puede utilizarlo para sintetizar un nuevo comportamiento que se adapte a los nuevos requerimientos.

En el caso abordado en (Bongard, Zykov y Lipson en 2006), el algoritmo de auto-modelado consta de cuatro etapas principales: 1) Ejecución de movimientos arbitrarios y registro de datos entregados por los sensores del robot. 2) Búsqueda de un auto-modelo sobre un grupo de modelos candidatos. 3) Síntesis de un nuevo comportamiento (controlador). 4) Ejecución del nuevo comportamiento y recolección de nuevos datos de los sensores.

En la etapa de búsqueda de auto-modelos, se utilizan datos recolectados por los sensores del robot real para compararlos con los datos de los robots candidatos. La información de los robots candidatos se recolecta en un entorno de simulación física.

La búsqueda de auto-modelos puede dividirse en las siguientes etapas: 1) Creación de un conjunto de candidatos. 2) Evaluación y selección de los candidatos. 3) Síntesis de nuevos movimientos exploratorios.

Un ciclo continuo de retro-alimentación, entre datos de los sensores del robot real y la información recolectada en las simulaciones físicas, permite al robot actualizar continuamente su auto-modelo y su comportamiento. Con esto adquiere la habilidad de adaptarse a cambios inesperados en su morfología o incluso en su entorno.

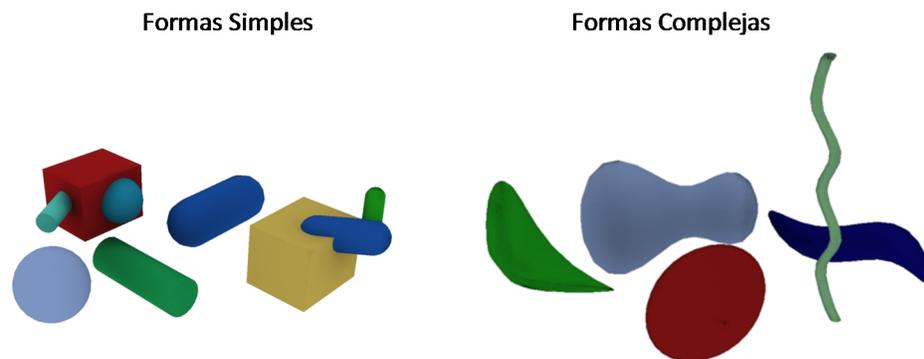
Dentro de la etapa de búsqueda de un auto-modelo, es muy importante la utilización de una métrica para la comparación de las señales. Esta métrica debe ser capaz de cuantificar cuan parecido es un modelo candidato al robot real, y esto solo utilizando la información recogida por los sensores.

En trabajos anteriores, como en (Bongard, Zykov y Lipson, 2006) y (Bongard y Lipson, 2004) se han implementado con éxito algoritmos de auto-modelado utilizando las métricas Euclidiana y Rolling Mean Metric.

Sin embargo estas métricas tienen algunas limitaciones en cuanto a robustez. En particular son muy sensibles a pequeños cambios en la posición inicial de un robot, así como a pequeños desfases entre las señales a comparar.

En este trabajo se realiza un estudio comparativo de las métricas existentes y se presenta una nueva métrica llamada Discriminante de Kullback, de naturaleza estadística, que muestra buenas propiedades en cuanto a la robustez en la comparación de señales.

Otra limitación de los trabajos realizados hasta el momento en auto-modelado como en (Bongard, Zykov y Lipson, 2006), (Zagaly Ruiz-del-Solar, 2004), (Bongard y Lipson, 2004), tiene que ver con los espacios de búsqueda utilizados. En particular en cuanto a las características geométricas de partes individuales de un robot, donde solo se ha utilizado formas simples: Cubos, cilindros, esferas o capsulas.



**Figura 1. Esquemmatización de formas simples y complejas.**

En este trabajo se presenta una nueva representación genérica de robots, basada en un sólido amorfo definido por 6 puntos de control, que son interpolados a través de una superficie de interpolación con splines cúbicos. Esta nueva representación amplía considerablemente el espacio de búsqueda incorporando geometrías complejas de partes individuales de un robot.

## **1.2 *Objetivos generales.***

Mejorar la flexibilidad del algoritmo de auto-modelado en cuanto al espacio de búsqueda de auto-modelos, con el objetivo que el algoritmo desarrollado pueda ser implementado en robots con variada morfología, incrementando por ende el campo de acción del algoritmo.

Además se busca proponer una nueva métrica, de naturaleza estadística, que sea robusta a la hora de trabajar con señales desfasadas u obtenidas de robots sobre los cuales existe cierta incertidumbre acerca de su posición inicial.

## **1.3 *Objetivos Específicos***

- Implementar un modelo de simulación física del robot Trepén.
- Utilizar el modelo del robot trepen para estudiar la eficacia de las métricas conocidas y proponer una nueva métrica capaz de mejorar la flexibilidad del algoritmo de auto- modelado.
- Implementar un algoritmo de auto-modelado en simulador del robot Trepén incorporando la nueva métrica propuesta y evaluar su eficacia.
- Estudiar nuevas posibles representaciones genéricas de robots.
- Evaluar la eficacia del algoritmo de auto modelado incorporando la nueva representación en robot con morfología distinta al robot Trepén

## 2 Antecedentes

### 2.1 Auto-Modelado

#### 2.1.1 Descripción

El algoritmo de auto-modelado otorga a un robot la capacidad de inferir características de su morfología a partir de mediciones internas acerca de su comportamiento, por ejemplo relaciones sensor-motrices.

El proceso comienza cuando el robot realiza ciertos movimientos accionando sus motores, luego recopila datos de los sensores ubicados en su cuerpo acerca de los movimientos realizados, posteriormente selecciona los auto-modelos que son más consistentes con las señales recolectadas. El proceso se repite continuamente recolectando información y mejorando el ajuste de los modelos.

#### 2.1.2 Etapas del auto-modelado

Movimientos exploratorios:

El robot acciona sus motores, con patrones que pueden ser al azar (en la primera iteración del algoritmo) o con patrones obtenidos anteriormente, con el fin de desambiguar ciertas discrepancias entre el modelo y la información recolectada.

Recolección de información:

Se utilizan los sensores instalados en el cuerpo del robot para recolectar series de tiempo durante la ejecución de algún comportamiento (controlador). Estos sensores pueden ser de variada naturaleza, como por ejemplo: Acelerómetros, sensores de velocidad, sensores de inclinación, sensores de presión, dinamómetros, voltímetros, amperímetros, etc.

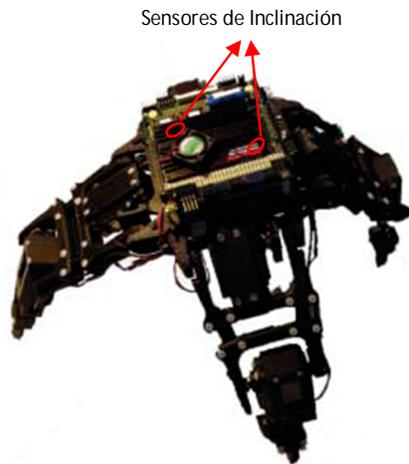
Para obtener las series de tiempo de los robots candidatos se utiliza un motor de simulación física, donde los robots candidatos ejecutan el mismo controlador del robot real y se registran los datos obtenidos con los sensores implementados en la simulación.

En el caso de (Bongard, Zykov y Lipson, 2006), se utilizaron dos sensores de inclinación ubicados en el cuerpo central de un robot cuadrúpedo, ver figura 2. Estos dos sensores fueron suficientes para realizar las comparaciones entre el robot físico y el modelo.

Definición del espacio de búsqueda:

En esta etapa usualmente se considera que la morfología de un robot cualquiera puede describirse en términos de su topología y geometría. La geometría se refiere a la forma de las partes individuales que componen al robot, mientras que la topología se refiere a las relaciones que existen entre estas partes. Los parámetros que describen la topología y geometría corresponden al espacio de búsqueda de auto-modelos de un robot cualquiera.

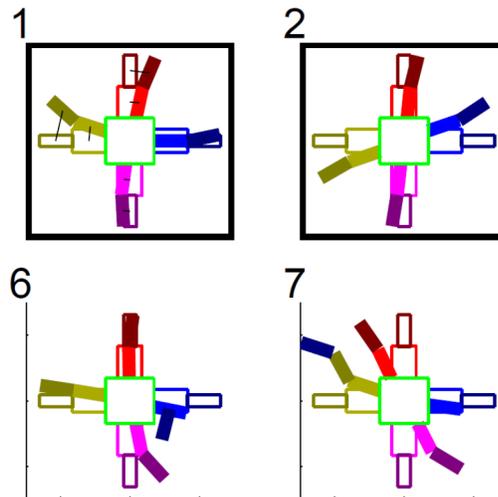
Un ejemplo de espacio de búsqueda se presenta en (Bongard, Zykov y Lipson, 2006), para el auto-modelado de un robot cuadrúpedo, Ver Figura 2.



**Figura 2. Robot cuadrúpedo Starfish, empleado en (Bongard, Zykov y Lipson, 2006).**

Este robot presenta 9 elementos: un cuerpo central cubico conectado a cuatro extremidades, cada una de ellas compuesta por dos cuerpos unidos por una articulación de bisagra, por lo que el robot posee ocho grados de libertad.

En este caso el espacio de búsqueda corresponde a dos parámetros para cada cuerpo del robot, el primer parámetro indica a que otro cuerpo está conectado y el segundo parámetro indica en que parte de su perímetro está ubicada la articulación. La Figura 3 muestra un ejemplo de espacios de búsqueda utilizados en (Bongard, Zykov y Lipson, 2006).



**Figura 3. Esquema de espacios de búsqueda de modelos empleado en Bongard, Zykov y Lipson, 2006).**

Para la búsqueda en este caso se generó un conjunto de 15 candidatos cuyos parámetros fueron elegidos al azar dentro del espacio de búsqueda.

Cabe destacar que en este trabajo no se consideraron dentro del espacio de búsqueda características como: La geometría de cada cuerpo, masa, matriz de inercia, coeficientes de roce, etc.

### Búsqueda del mejor postor:

Una vez que está definido el espacio de búsqueda de modelos, se genera un conjunto de candidatos a ser el modelo elegido.

Entonces se debe establecer un criterio para calificar la capacidad de un modelo de reproducir los datos recogidos del robot real. Esta calificación es una medida del desacuerdo entre el comportamiento del modelo (resultante al ejecutar un controlador en un modelo simulado) y los resultados experimentales. Por ejemplo se puede utilizar como criterio la norma euclidiana del error entre las series de tiempo experimentales y las del modelo.

Una dificultad intrínseca al proceso de comparar las series de tiempo con los resultados de la simulación es que al tratarse de sistemas no lineales altamente acoplados, una pequeña discrepancia presente al comienzo de la simulación puede resultar en discrepancias muy grandes al final de la simulación (Bongard y Lipson, 2004).

Debido a esta dificultad para identificar el ajuste de los modelos, adquiere vital importancia la definición de una métrica que pueda resolver eficazmente el problema de calificar el ajuste de un modelo y compararlo con los demás.

La norma utilizada en el trabajo descrito anteriormente (Bongard y Lipson, 2004) es *Rolling mean metric*. Sin embargo se han utilizado otras métricas (Zagal et al, 2009), (Zagal y Ruiz-del-Solar, 2008), donde se utiliza la norma llamada *Experience Metric*, que utiliza parámetros estadísticos incorporando información sobre el tiempo de ciclo de la experiencia.

El proceso de evaluación y selección de un conjunto de candidatos es manejado por un algoritmo de búsqueda, en (Bongard, Zykov y Lipson en 2006) se utilizan algoritmos genéticos para realizar la búsqueda del mejor candidato.

### Síntesis de nuevos movimientos exploratorios:

Cuando un robot ejecuta un controlador particular, su comportamiento va a depender solo de ciertos parámetros de su morfología que son relevantes al ejecutar ese controlador. Por lo tanto la selección de un auto-modelo está restringida solo a aquellos parámetros relevantes.

Una posibilidad para lidiar con este problema es ejecutar muchos controladores diferentes con tal de rescatar la mayor cantidad de parámetros relevantes. Este conjunto de controladores puede ser generado al azar.

Sin embargo en (Bongard, Zykov y Lipson en 2006) se plantea otra forma de generar este conjunto de controladores. Se propone utilizar un principio de retro-alimentación entre el conjunto de candidatos y el conjunto de controladores. Específicamente se plantea una búsqueda de nuevos controladores que maximicen las discrepancias entre los datos del auto-modelo seleccionado y los datos del robot real. Se ha mostrado que utilizando este principio se puede acelerar el proceso de búsqueda de auto modelos.

Entonces se trata de dos conjuntos que se retro-alimentan de la siguiente forma:

- El primer conjunto corresponde a los modelos candidatos para la estructura del robot real. Se realiza una selección sobre este conjunto utilizando el criterio de minimizar las discrepancias con los resultados experimentales.

- El segundo conjunto corresponde a las pruebas posibles a realizar en la siguiente iteración. Se realiza una selección sobre este conjunto utilizando el criterio de maximizar las discrepancias del modelo elegido con respecto a los resultados experimentales.

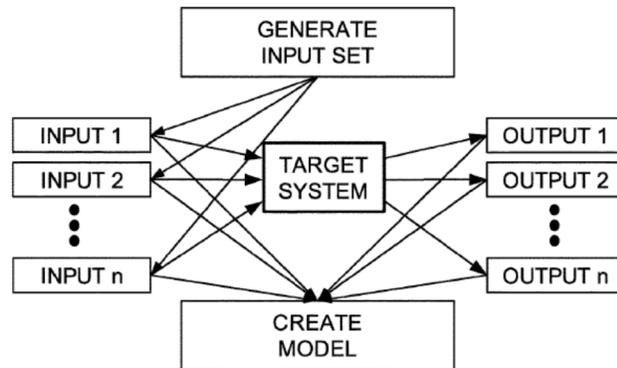


Figura 4. Esquema del funcionamiento del algoritmo de co-evolución entre experimento y modelo descrito en (Bongard y Lipson, 2005).

#### Recolección de nueva información:

Se ejecutan los nuevos controladores sintetizados en la etapa anterior para recolectar nuevas series de tiempo, y así volver a mejorar el ajuste de los modelos.

### 2.1.3 Síntesis de Movimiento

Un requisito fundamental para que un robot sea capaz de adaptarse a cambios inesperados en su morfología es que logre sintetizar movimientos que se adapten a su nueva condición, este proceso puede corresponder a una modificación de un movimiento existente o a la creación de uno completamente nuevo.

En ese sentido, interesa que dado un objetivo impuesto, por ejemplo desplazarse, el robot logre sintetizar una secuencia de posiciones en el tiempo para cada una de sus articulaciones y así aproximarse al cumplimiento del objetivo impuesto.

Para esto nuevamente se debe acotar el espacio de búsqueda de soluciones, por ejemplo puede imponerse que la posición de una articulación en función del tiempo siga una senoide, con ciertos parámetros indeterminados. Estos parámetros representan el espacio de búsqueda para el algoritmo.

Entonces en esta etapa nuevamente se genera una población de soluciones, dentro del espacio de búsqueda, que compiten por cumplir el objetivo impuesto.

En (Zagal y Ruiz-del-Solar, 2008) se expone la implementación de un algoritmo de síntesis de movimiento para un hexápodo.

En este caso se utilizó un controlador del movimiento de cada articulación (ángulo) basado en una senoide que oscila alrededor de un ángulo de referencia.

$$r_i = \theta_i + a_i \sin(\omega t + \varphi_i) \quad (1)$$

Dónde:

$\theta_i$  = Ángulo de referencia para la articulación  $i$ .

$a_i$  = Amplitud de movimiento para articulación  $i$ .

$\omega$  = Frecuencia de 60 Hz seleccionada a priori en esta experiencia.

$\varphi_i$  = Ángulo de fase para articulación  $i$ .

Entonces dado este controlador, se define el espacio de búsqueda como los parámetros  $a_i, \varphi_i$ .

### 2.1.4 Métricas de comparación

En el contexto de un algoritmo de auto modelado una métrica cumple la función de comparar señales o comportamientos obtenidos de un robot (real o simulado) y entregar un valor que corresponde a una evaluación de cuan parecidos son dos robots.

Una buena métrica es capaz de entregar una buena evaluación al comparar robots parecidos y una mala evaluación al comparar robots muy distintos. Osea se espera que para un pequeño cambio en la morfología de un robot, la evaluación sufra también un pequeño cambio.

Además, para una graduación de cambios en algún parámetro del robot, se espera que la métrica entregue una función monótona al evaluar cambios graduales en los modelos.

### 2.1.5 Métrica Euclidiana

Para el caso de comparación de series de tiempo, esta métrica calcula la suma del error entre dos señales, además suma el error para distintos controladores ejecutados en el robot y para los distintos sensores ubicados en este. Esta norma fue utilizada en (J.C. Bongard y H. Lipson, 2004)

La definición formal de esta métrica es:

$$f(g_p) = \frac{\sum_{i=1}^c \sum_{k=0}^h |S_{phy}^i(t) - S_{sim}^i(t)|}{ch} \quad (2)$$

Dónde:

$f$ : Es la evaluación o fitness para la comparación de dos robots.

$g_p$ : Genotipo o características de un robot candidato  $p$ .

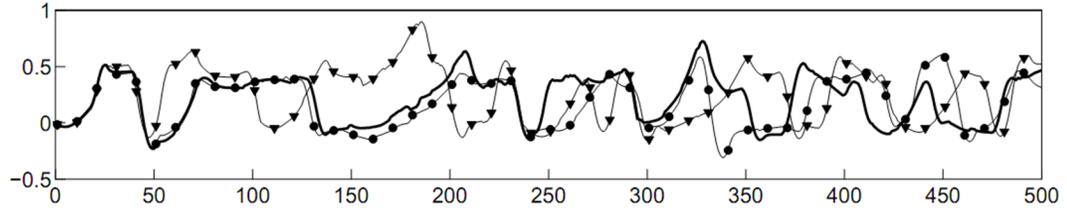
$c$ : Numero de controladores.

$h$ : Largo de la serie de tiempo.

$S_{phy}^i(t)$ : Serie de tiempo para controlador  $i$ .

### 2.1.6 Métrica de Promedio Rodante (Rolling Mean Metric)

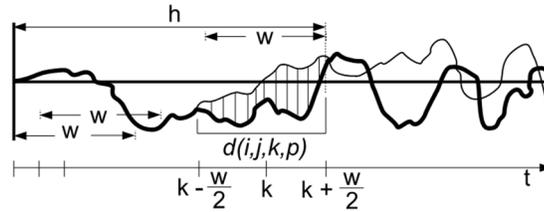
Esta métrica presentada en (J.C. Bongard y H. Lipson, 2004a) se hace cargo de la problemática de analizar señales provenientes de sistemas no-lineales altamente acoplados. En la figura 5 se muestran tres series de tiempo provenientes de robots con pequeñas variaciones en su morfología. Puede observarse que luego de un tiempo específico ( $t=50$ ) las señales divergen y se pierde la correlación entre las señales.



**Figura 5. Tres series de tiempo obtenidas de robot cuyo servo motor fue debilitado en un 10% (línea gruesa), 20% (círculos) y 50% (triángulos). (J.C. Bongard y H. Lipson, 2004).**

La métrica de promedio rodante se hace cargo de este fenómeno, limitando la comparación a una ventana temporal pequeña al comienzo de la simulación. Además incorpora un filtro de suavizado de la señal manejado por un parámetro  $w$ .

Entonces se realiza una suma del error sobre el tiempo, los controladores y los sensores. La figura 6 esquematiza el funcionamiento de la métrica y el filtro de suavizado.



**Figura 6. Esquema explicativo del funcionamiento de la Rolling Mean Metric.(J.C. Bongard y H. Lipson, 2004).**

La definición formal de la métrica es:

$$f(g_p) = \frac{\sum_{i=1}^c \sum_{j=1}^n \sum_{k=w/2}^{h-w/2} d(i, j, k, p)}{cn(h-w)} \quad (3)$$

$$d(i, j, k, p) = \frac{\sum_{t=k-w/2}^{t=k+w/2} |S_{phy}^{(i,j)}(t) - S_{sim}^{(i,j,p)}(t)|}{ch} \quad (4)$$

Dónde:

$f$ : Es la evaluación o fitness para la comparación de dos robots.

$g_p$ : Genotipo o características de un robot candidato  $p$ .

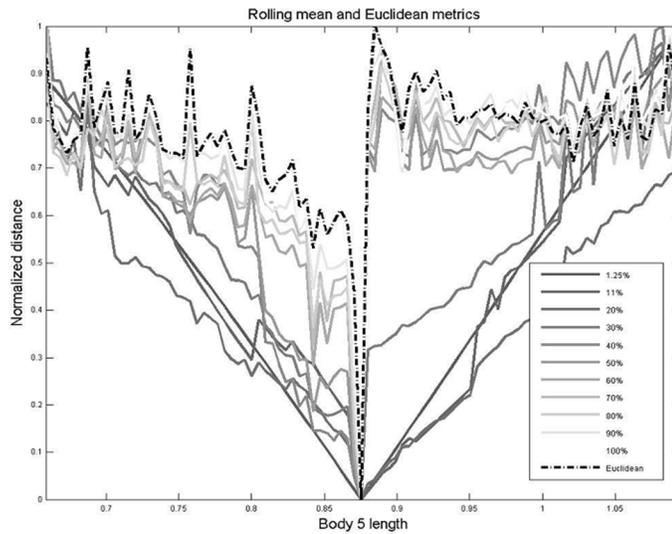
$c$ : Numero de controladores.

$h$ : Largo de la ventana temporal.

$S_{Phy}^{(i,j,p)}(t)$ : Serie de tiempo para controlador  $i$ , sensor  $j$  y modelo  $p$ .

$w$ : Largo del intervalo del filtro de suavizado.

En la figura 7 se muestra cómo cambia la monotonía en la evaluación al variar el parámetro  $h$ , es decir el largo de la ventana temporal.



**Figura 7.** Se muestra el patrón de evaluación de la métrica euclidiana y la Rolling Mean Metric variando el parámetro  $h$  en un rango del 1.25% a un 100% del largo de la serie de tiempo. (Zagal y Ruiz-del-Solar, 2008).

## 2.2 Representación

Para el funcionamiento del algoritmo de auto modelado es necesario diseñar una representación paramétrica de las características del robot. A su vez esta representación debe ser compatible con una codificación aplicable a la optimización con algoritmos genéticos.

### 2.2.1 Topología

Se refiere a las relaciones existentes entre las partes de un robot, por ejemplo entre dos cuerpos puede existir una unión tipo bisagra, que restringe el movimiento relativo de los cuerpos a un grado de libertad.

Algunos ejemplos de relaciones topológicas compatibles con la Libertia Open Dynamics Engine se muestran en la figura 8.

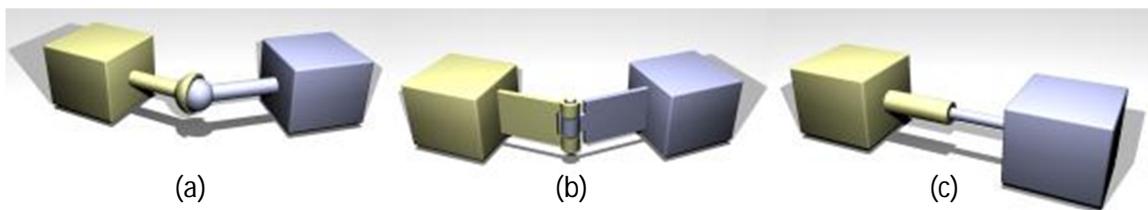


Figura 8. Ejemplos de articulaciones: Universal (a), Bisagra (b), Pistón (c). (Open Dynamics Engine: reference Manual).

En (Bongard, Zykov y Lipson, 2006) se utilizó una representación topológica para un robot cuadrúpedo basada en dos parámetros para cada cuerpo; el primero describe con que otro cuerpo está relacionado y el segundo indica en que parte de su perímetro se encuentra la articulación. En la figura 9 se esquematizan las variaciones que se pueden obtener con esta representación.

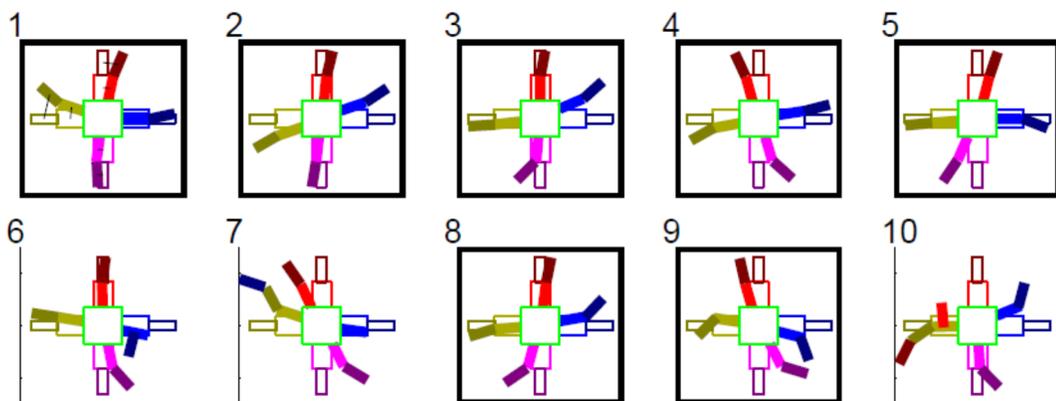


Figura 9. Esquema de tipos de relaciones topológicas empleado en (Bongard, Zykov y Lipson, 2006).

En (Zagaly Ruiz-del-Solar, 2008) para la representación de la topología de un hexápodo se utilizaron articulaciones de bisagra. En la figura 10 se muestra como estas articulaciones se situaron variando el eje de rotación de la bisagra.

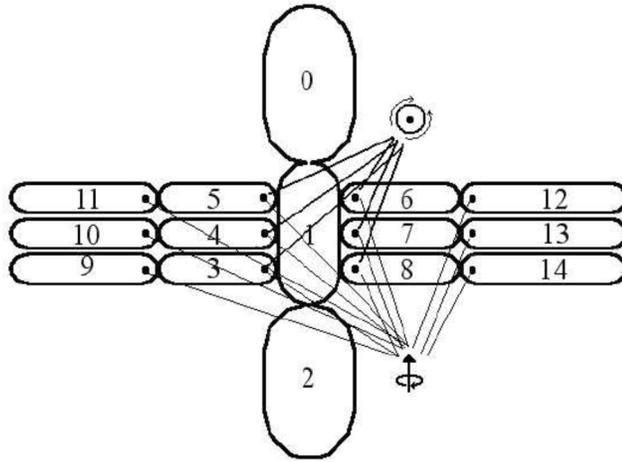


Figura 10. Esquema relaciones topológicas de un hexápodo utilizado en (Zagal y Ruiz-del-Solar, 2008).

## 2.2.2 Geometría

En (Bongard, Zykov y Lipson, 2006), (Zagaly Ruiz-del-Solar, 2004), (Bongard y Lipson, 2004), se utilizan formas geométricas simples para representar partes individuales de un robot. En particular se utilizan formas como: Cubos, cilindros, esferas o capsulas.

La virtud de utilizar formas geométricas simples radica en que su representación paramétrica es muy económica, o sea se requieren muy pocos parámetros para la descripción del objeto.

En el caso de (Bongard, Zykov y Lipson, 2006) se representó la geometría de un robot cuadrúpedo utilizando cilindros y cajas para cada parte del cuerpo. La parametrización se definió mediante variaciones discretas de las dimensiones de dichas cajas y cilindros.

En (Zagaly Ruiz-del-Solar, 2008), se realizó la representación geométrica de un hexápodo utilizando capsulas con distintas dimensiones. Estas dimensiones se parametrizaron en valores discretos. La figura 11 muestra cómo se varió el largo de uno de los cuerpos del robot.

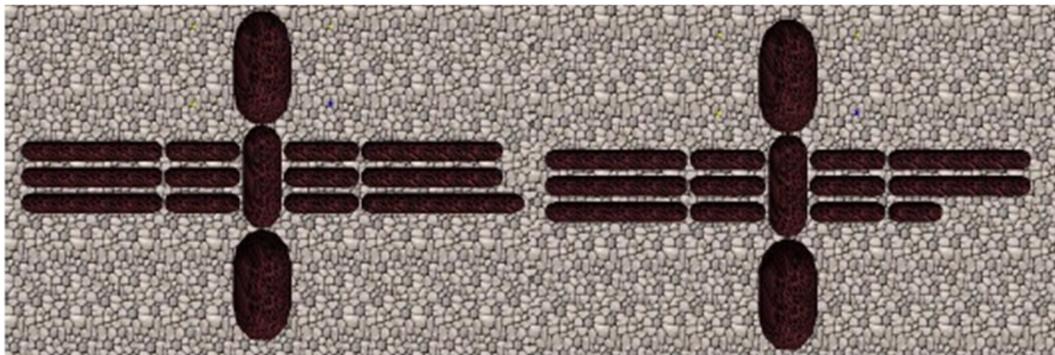


Figura 11. Esquema de variaciones geométricas de un hexápodo utilizado en(Zagal y Ruiz-del-Solar, 2008).

## 2.3 Representación a través de splines cúbicos

### 2.3.1 Splines

Spline es una curva definida en porciones mediante polinomios. En los problemas de interpolación, se utiliza a menudo la interpolación mediante splines con el fin de utilizar polinomios de bajo grado, evitando así las oscilaciones, indeseables en la mayoría de las aplicaciones, encontradas al interpolar mediante polinomios de grado elevado.

Para el ajuste de curvas, los splines se utilizan para aproximar formas complicadas. La simplicidad de la representación y la facilidad de cómputo de los splines los hacen comunes para la representación de curvas en informática, particularmente en el terreno de los gráficos por ordenador.

Los splines cúbicos son un caso particular de interpolación por tramos donde el grado de los polinomios utilizados es 3.

En notación matricial un spline de interpolación cubico entre dos puntos se escribe de la siguiente manera.

$$P(x) = [x^3 x^2 x 1] \begin{bmatrix} a_3 \\ a_2 \\ a_1 \\ a_0 \end{bmatrix} \quad (5)$$

Donde los coeficientes  $a_i$  representan valores reales que caracterizan la curva de interpolación.

### 2.3.2 Interpolación de Hermite

Para el caso particular de una interpolación entre dos puntos e incluyendo como restricción la pendiente en los extremos de la curva, figura 12, puede usarse la notación matricial de interpolación de Hermite para obtener los coeficientes de interpolación  $a_i$ .

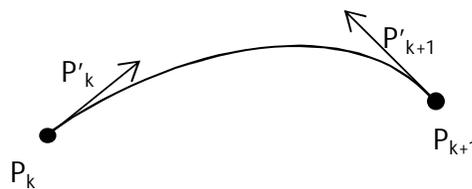


Figura 12. Interpolación por spline de Hermite entre dos puntos.

$$P(x) = [x^3 x^2 x 1] \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} P(0) \\ P(1) \\ P'(0) \\ P'(1) \end{bmatrix} \quad (6)$$

Donde:

$P(0)$ : representa la evaluación del polinomio en el primer punto.

$P(1)$ : representa la evaluación del polinomio en el segundo punto.

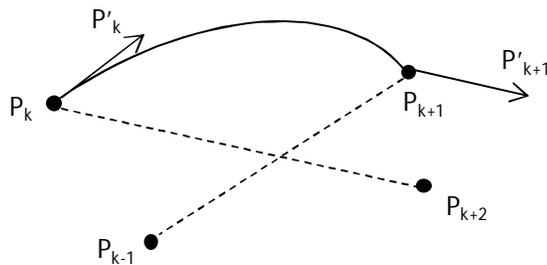
$P'(0)$ : representa la evaluación de la derivada del polinomio en el primer punto.

$P'(1)$ : representa la evaluación de la derivada del polinomio en el segundo punto.

### 2.3.3 Splines cardinales

En el caso de la interpolación de Hermite es necesario controlar la pendiente de la curva en cada par de puntos, por lo tanto es un mecanismo poco práctico cuando se quiere interpolar una serie arbitraria de puntos.

La figura 13 muestra una serie de 4 puntos consecutivos que pueden ser parte un conjunto arbitrario de puntos, en este caso para realizar una interpolación entre los dos puntos centrales, puede realizarse una estimación de la pendiente en sus extremos, utilizando información de los puntos extremos.



**Figura 13. Interpolación por spline cardinales utilizando series de 4 puntos.**

Las pendientes en el punto  $k$  y  $k+1$  quedan respectivamente:

$$\begin{aligned} P'_k &= \frac{1-t}{2} (P_{k+1} - P_{k-1}) \\ P'_{k+1} &= \frac{1-t}{2} (P_{k+2} - P_k) \end{aligned} \quad (7)$$

Donde:

$t$ : Parámetro de tensión para la pendiente.

## 2.4 Algoritmos genéticos

Los algoritmos genéticos son métodos de búsqueda u optimización que se basan en el mecanismo de selección natural.

En una analogía directa trabajan manejando una población de individuos donde cada uno corresponde a una solución factible de un problema.

A cada uno de estos individuos se le asigna una evaluación de acuerdo a que tan bien satisfacen algún criterio de selección. Entonces la idea es que los mejores individuos (mejor evaluados) traspasen sus características a la siguiente generación.

El proceso de traspaso de información de una generación a otra se rige por un operador de selección, que utiliza las evaluaciones de cada individuo para seleccionar según algún criterio cuáles podrán traspasar sus características.

Una vez que se han seleccionado los mejores individuos de una población se utiliza el operador cruce o crossover para recombinar las características de los individuos seleccionados en nuevos individuos de la próxima generación.

El proceso repetitivo de traspasar las características de los mejores individuos de una generación a otra otorga al algoritmo la capacidad de generar nuevas generaciones con individuos mejor evaluados que las generaciones anteriores. Sin embargo esta convergencia de las poblaciones hacia una solución óptima no está asegurada.

### 2.4.1 Individuos y Representación

En el contexto de un algoritmo de búsqueda u optimización un individuo corresponde a una solución factible para el problema dado.

Sin embargo para que un individuo sea parte de un algoritmo de búsqueda genética es necesario establecer una representación paramétrica de éste, esta representación para un individuo en particular se llama cromosoma y cada parámetro de éste se le llama gen.

Algunas representaciones utilizadas son:

- Binaria: Los genes dentro de un cromosoma pueden tomar valores de cero o uno.
- Entera: Los genes dentro de un cromosoma pueden tomar solo valores enteros.
- Real: Los genes dentro de un cromosoma pueden tomar cualquier valor real.

### 2.4.2 Población

Una población es simplemente un grupo de individuos pertenecientes a una misma generación. Las poblaciones evolucionan a través de las generaciones.

Cada generación es producto de la acción del operador de selección y cruzamiento aplicado a la generación anterior. Para el caso de la población inicial ésta generalmente se crea de manera aleatoria.

### 2.4.3 Selección

El operador selección cumple la función de traspasar de una generación a otra las mejores características de los individuos. Hay muchas formas de realizar esta selección, pero todas requieren la definición de una función objetivo que entrega una evaluación de la calidad de los individuos de una población. Esta evaluación es específica para cada problema de búsqueda u optimización. Algunos de los mecanismos de selección utilizados son:

- Ruleta: La probabilidad de que un individuo traspase sus genes a la siguiente generación es directamente proporcional a la evaluación que reciba de la función objetivo. Es equivalente a una ruleta donde el área asociada a cada individuo es proporcional a la evaluación que este recibe.
- Rango: En este caso los individuos se clasifican en rangos de evaluación asignándoseles un valor según su rango pero que es distinto al valor de su evaluación. Lugo se realiza una selección otorgando una probabilidad de selección a cada individuo que es proporcional a su rango.
- Torneo: Se crean sub grupos de la población total y se elige el mejor representante de cada grupo.

### 2.4.4 Cruce o Crossover

Es el mecanismo mediante el cual se combinan dos individuos para formar uno nuevo. Algunos cruces utilizados son:

- Cruce puntual: Se selecciona un punto de los cromosomas progenitores y se intercambian los genes a la izquierda o derecha de ese punto.
- Cruce n-puntos: Se eligen varios puntos al azar en los cromosomas de los progenitores y se intercambian los genes entre esos puntos.
- Cruce uniforme: se crea un patrón aleatorio de ceros y unos del mismo largo que los progenitores y se intercambian los genes donde aparece un uno.

### 2.4.5 Mutación

El objetivo de la mutación es aumentar la variabilidad de las nuevas poblaciones y funciona incluyendo algún cambio aleatorio en el cromosoma de algún nuevo individuo. En un algoritmo genético se debe establecer la probabilidad o frecuencia con que se producen mutaciones.

## 2.5 Discriminante de Kullback

En el contexto de las probabilidades el Discriminante de Kullback es un indicador de similitud entre dos distribuciones de probabilidad P y Q. Dentro de la teoría de la información se la conoce como divergencia de la Información, ganancia de la Información o Entropía Relativa.

Es ampliamente utilizada en la identificación de modelos mediante el procesamiento de señales, al entregar una medida de cuan probable es que dos modelos sean iguales en función de la distribución que siguen los datos empíricos entregados por cada uno de ellos (Patricio Parada y Juan Cristóbal Zagal, 2010)

El Discriminante de Kullback en rigor no es una distancia, ya que no es simétrica, es decir no se cumple que  $D_{KL}(P//Q) \neq D_{KL}(Q//P)$ , sin embargo si establece una medida de probabilidad de que dos distribuciones empíricas obedezcan al mismo modelo.

### 2.5.1 Definición para distribuciones continuas

Para P y Q distribuciones de una variable aleatoria continua, el Discriminante de Kullback se define como:

$$D_{KL}(P//Q) = \int_{-\infty}^{\infty} p(x) \log \left( \frac{p(x)}{q(x)} \right) dx \quad (8)$$

Donde p y q denotan las densidades de probabilidad de P y Q.

### 2.5.2 Definición para distribuciones discretas (histogramas)

Para P y Q distribuciones de probabilidad de una variable aleatoria discreta, el Discriminante de Kullback se define como:

$$D_{KL}(P//Q) = \sum_i P(i) \log \left( \frac{P(i)}{Q(i)} \right) \quad (9)$$

Donde p y q denotan las densidades de probabilidad de P y Q.

### 2.5.3 Propiedades

- Es siempre positiva (puede probarse usando la desigualdad de Jensen).
- Es nula si y sólo si  $P = Q$ .
- No es simétrica (por lo que no se trata de una distancia)

## 2.6 Motor de Simulación física

Un motor de simulación física es un paquete de software que permite la implementación de un entorno de simulación de sistemas dinámicos, donde se pueden incorporar múltiples sólidos rígidos articulados interactuando en el tiempo, bajo el efecto de ciertas fuerzas y restricciones.

Además realiza una simulación de las colisiones elásticas entre los cuerpos, así como de las fuerzas de roce que se producen entre estos.

Algunos motores de simulación física existentes en el mercado son:

(Motor)	(Licencia)	(Costo: Edu/Com)	(Plataforma: PC)	(Plataforma: Consola)
Engine	License	Cost : Edu/Com	Platform:PC	Platform:Console
AGEIA PhysX / Novodex	EULA	Free/Free	Win32/Linux/-	Xbox360/PS3/-
Bullet	Open, Zlib	Free/Free	Win32/Linux/Mac	Xbox360/PS3/-
JigLib	Open	Free/Free	Win32/Linux/-	-/-/-
Newton	EULA	Free/Free	Win32/Linux/Mac	-/-/-
Open Dynamics Engine	Open, LGPL/BSD	Free/Free	Win32/Linux/Mac	Xbox360/PS3/PSP
Tokamak	Open, BSD	Free/Free	Win32/Linux/-	-/-/-
True Axis	EULA	Free/Hobby+Full	Win32/Linux/Mac	Xbox360/-/-

Figura 14. Tabla comparativa sobre distintos motores de simulación (Adrian Boeing Thomas Bräunl 2008).

### 2.6.1 Open Dynamics Engine (ODE)

Es un motor de simulación física que funciona como una librería para c++, incorpora manejo de sistemas de sólidos rígidos, detección de colisiones, respuesta a colisiones y modelo de roce.

En cuanto a la detección de colisiones la ODE soporta colisiones de cuerpos simples y objetos compuestos por malla de triángulos.

### 3 Metodología General de Trabajo

El método de trabajo que se desarrolló se basa principalmente en utilizar un entorno de simulación de sistemas físicos, para estudiar el desempeño de nuevos métodos y aplicaciones del algoritmo de auto-modelado.

En el entorno de simulación física se incluyen modelos de robots, con los cuales se pueden realizar los experimentos. Las experiencias realizadas se pueden dividir en general en los siguientes pasos:

- Implementar una simulación para un robot en particular.
- Crear un robot que se considera como el robot real.
- Crear una población de robots candidatos.
- Instalar sensores en los robots.
- Ejecutar algún comportamiento en los robots (controlador).
- Recolectar información de los sensores.
- Analizar la información en el contexto de un algoritmo de auto-modelado.

En cuanto al entorno de simulación, sus parámetros físicos permanecen fijos para las distintas simulaciones, sin embargo las características de los robots son propias de cada experimento.

Los algoritmos de búsqueda de auto-modelos varían de una experiencia a otra en cuanto a la población de candidatos, número de generaciones y función objetivo.

Los resultados de los experimentos, tanto de estudio de métricas como búsqueda con algoritmos genéticos, se muestran en gráficos que buscan mostrar el desempeño de nuevos métodos, en el contexto de un algoritmo de auto-modelado.

## 4 Resultados

### 4.1 Modelo de Simulación

El modelo de simulación del robot Trepén se realizó utilizando la librería Open Dynamics Engine para C++. La programación se realizó utilizando Visual Basic Express.

#### 4.1.1 Parámetros de la simulación:

La tabla 1 muestra los parámetros utilizados en la simulación.

Sistema de Unidades	Sistema Internacional
Gravedad	9.8 [m/s <sup>2</sup> ]
Coefficiente de Fricción ( $\mu$ )	0.3
Paso temporal	0.02 [s]
Tipo de simulación	Determinista

Tabla 1. Tabla de parámetros de simulación.

#### 4.1.2 Modelo robot trepen:

En la figura 15 se muestra la estructura del robot implementado en la simulación, consta de 4 extremidades y cada una de estas constituidas por dos cuerpos iguales. Entre cada cuerpo existe una articulación tipo bisagra.

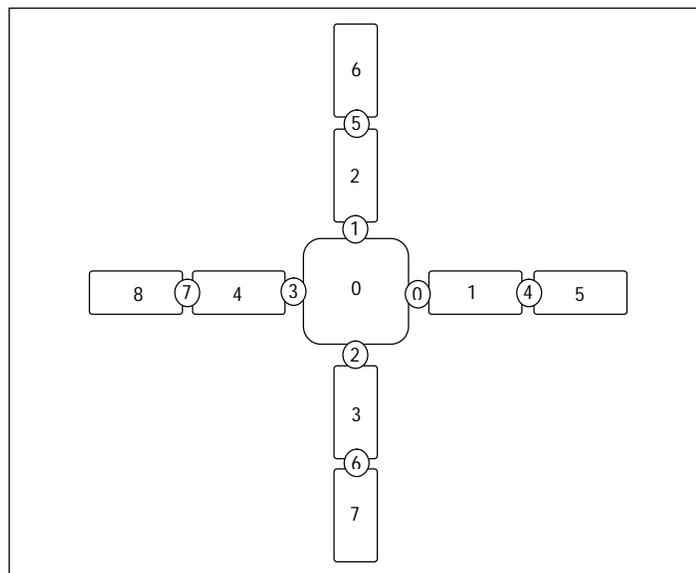


Figura 15. Esquema enumeración de cuerpos y articulaciones del robot simulado.

#### 4.1.3 Dimensiones

El robot cuenta con un cuerpo central de 0.05 x 0.2 x 0.2 m y 8 cuerpos en las extremidades con dimensiones 0.05 x 0.05 x 0.2 m.

#### 4.1.4 Topología

En la figura 15 se muestran las articulaciones de bisagra enumerados desde el 0 hasta el 7 (encerrados en círculos).

##### Servo-motores

En cada articulación del robot se implementa un servo motor simulado, este motor funciona imponiendo cierta velocidad angular a una articulación dada.

Para controlar cada articulación con una señal angular, es necesario implementar un controlador proporcional sobre el motor integrado en la ODE. La expresión que controla la velocidad impuesta en cada momento sobre una articulación es la siguiente.

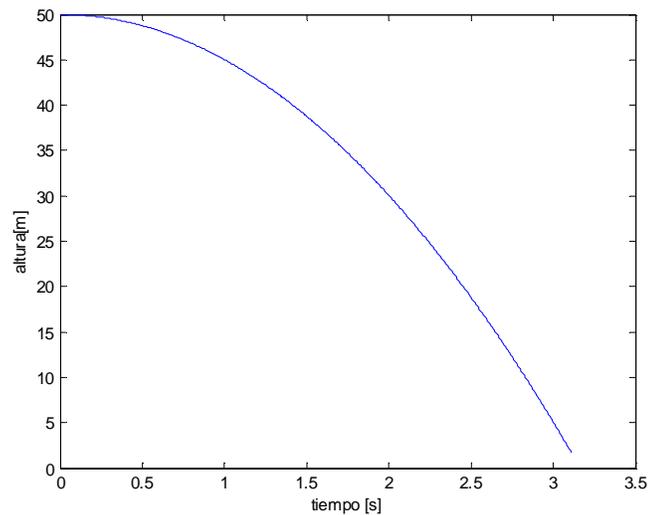
$$\omega(t) = -c (\text{Angulo actual} - \text{Angulo Objetivo}) \quad (10)$$

La constante proporcional se fijó en  $c = 15$ .

#### 4.1.5 Validación del modelo de simulación

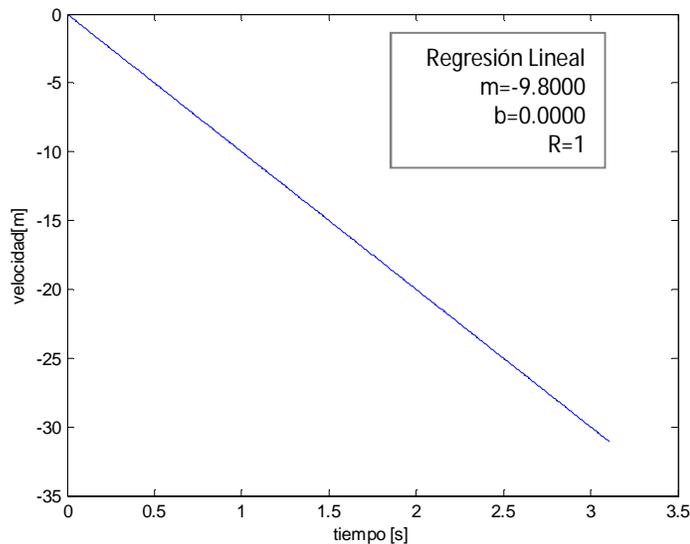
Para validar la precisión del modelo de simulación y asegurarse de que está trabajando en las unidades correspondientes se desarrolló un experimento que consiste en registrar la posición de un objeto en caída libre. En particular se trata de una esfera de radio  $r=0.1$  [m] y masa de 1 kg que se deja caer desde una altura de 50 [m]. Los parámetros del simulador son los descritos en la tabla N° 1.

Un sensor ubicado en el centro de masa del cuerpo indica la altura del objeto para cada tiempo. El resultado se muestra en la figura 16.



**Figura 16. Grafico altura en función del tiempo para una caída libre en el simulador.**

Realizando una derivación numérica de la posición con respecto al tiempo se obtiene la curva para la velocidad que se muestra en la figura 17. Al realizar una regresión lineal a la curva de velocidad se obtiene la pendiente de la curva  $m$  que corresponde a la aceleración de gravedad de  $9.8 \text{ m/s}^2$ .



**Figura 17. Velocidad del cuerpo en función del tiempo.**

## 4.2 Estudio de Métricas

Utilizando el modelo de simulación descrito anteriormente se diseñó una serie de experimentos con el objetivo de evaluar la eficacia de distintas métricas para detectar discrepancias entre distintos modelos del robot.

### 4.2.1 Detalle experimento N° 1

Para este experimento se utilizó el modelo del robot descrito en la sección 5.1. Se realizaron 160 variaciones del robot simulado. El parámetro que se varió en cada modelo fue el largo de una sola de las extremidades, que varía dentro del siguiente intervalo:  $L = [0.05m, 0.35m]$ . Este intervalo se dividió en 160 partes iguales. En la figura 18 se muestra esquemáticamente 10 modelos utilizados para la experiencia.

Para esta experiencia se definió como robot real al que presenta un largo de pata  $L=0.2$ , por lo que todas las métricas de comparación se ejecutaron tomando este robot como referencia.

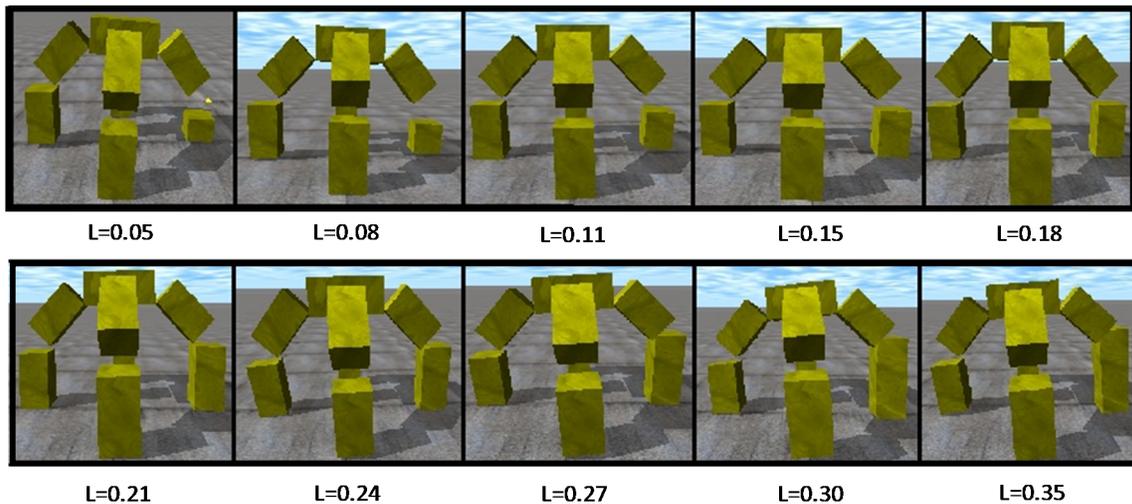


Figura 18. Descripción esquemática de la variación del parámetro durante el experimento.

## Sensores

Se implementaron un sensor de altura en el centro de masa del cuerpo central del robot. Este sensor registró información durante 60 segundos.

## Controladores

Durante las simulaciones, a cada articulación del robot trepen se le entregó una señal angular en función del tiempo. La señal que recibe cada articulación de detalla en la tabla N° 2.

Señal Angular [Rad]
A1: $f=0.8 \cdot \sin(1t)$
A2: $f=0.8 \cdot \sin(2t)$
A3: $f=0.8 \cdot \sin(3t)$
A4: $f=0.8 \cdot \sin(4t)$
A5: $f=0.8 \cdot \sin(5t)$
A6: $f=0.8 \cdot \sin(6t)$
A7: $f=0.8 \cdot \sin(7t)$
A8: $f=0.8 \cdot \sin(8t)$

**Tabla 2. Detalle de la señal angular que recibe cada articulación del robot.**

### 4.2.2 Series de tiempo

La figura 19 muestra las series de tiempo obtenidas para cada uno de los 160 candidatos y el robot real, cada modelo con un color distinto. Puede observarse que las señales se superponen en la mayoría de los puntos. Las mediciones se realizaron durante un tiempo de simulación de 60 [s], con esto se obtienen series de tiempo con 3000 puntos.

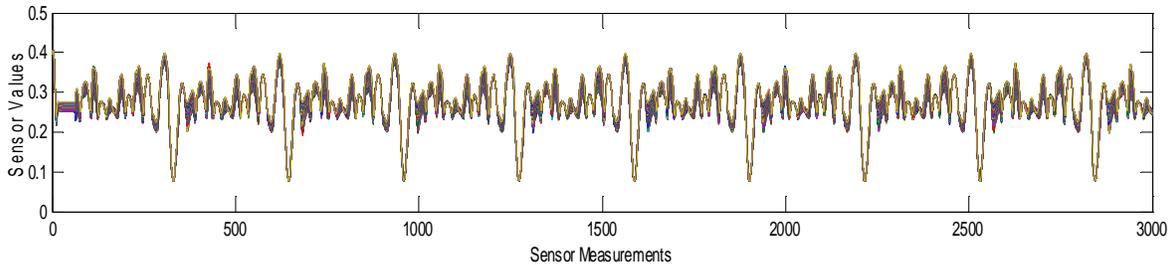


Figura 19. Series de tiempo de los 160 modelos candidatos y el modelo real.

### 4.2.3 Métrica euclidiana

La figura 20 muestra el patrón de evaluación que se obtiene aplicando la métrica euclidiana a los 160 modelos candidatos. Puede observarse que el valor de la evaluación crece a medida que el candidato presenta una mayor variación con respecto al modelo real, sin embargo no se observa perfecta gradualidad (monotonía) de la métrica, al evaluar pequeñas variaciones del largo de la pata con respecto al robot real.

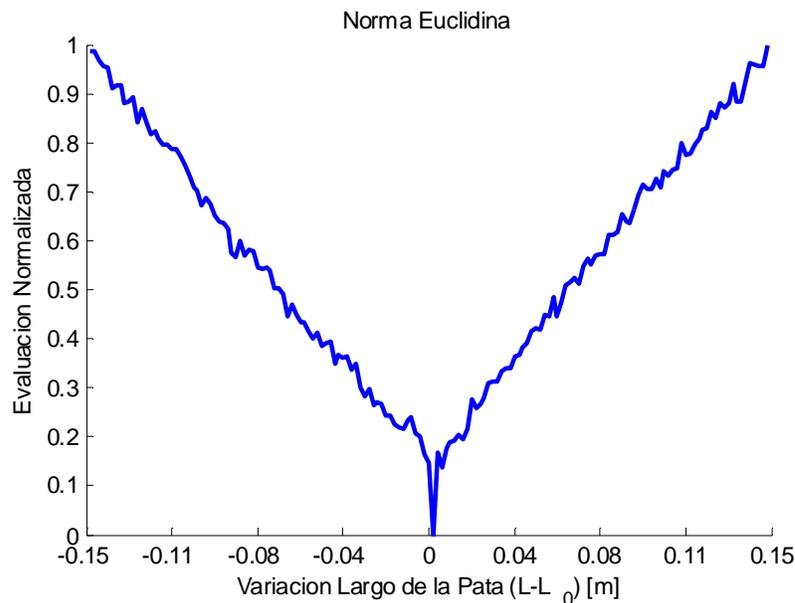


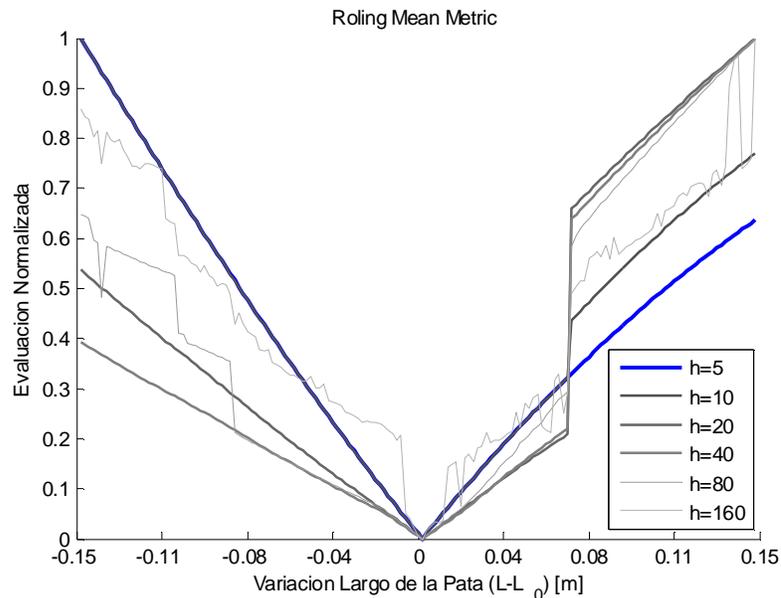
Figura 20. Evaluación métrica euclidiana para distintas variaciones del largo de la pata ( $L_0=0.2$  m).

#### 4.2.4 Métrica de promedio rodante (Rolling Mean Metric)

Esta métrica se caracteriza porque realiza la comparación entre señales dentro de una ventana temporal específica de largo  $h$ . Esta métrica además incorpora un filtro de suavizado controlado por un parámetro  $w$ . La expresión analítica de esta métrica se muestra en las ecuaciones N° 3 y N° 4.

Para esta experiencia se utilizó una ventana temporal  $h$  variable, mientras que el filtro de suavizado se fijó en  $w=2$ .

En la Figura 21 se muestran los patrones de evaluación obtenidos para distintas extensiones de la ventana temporal. Puede observarse que la monotonía de la evaluación mejora notoriamente al disminuir el tamaño de la ventana temporal.



**Figura 21. Patrón de evaluaciones para distintos largo de la ventana temporal  $h$ .**

Esta métrica es especialmente eficaz al utilizar una ventana temporal pequeña al inicio de la simulación. En cuanto al parámetro  $w$  que controla el filtro de suavizado, en la figura 22 se observa que no hay mayores cambios al variar el parámetro  $w$ . esto puede deberse a que al ser señales generadas en un simulador estas tienen un bajo componente de ruido, es por esto que este parámetro podría cobrar importancia cuando se analizan series de tiempo obtenidas con sensores reales que contengan algún componente de ruido.

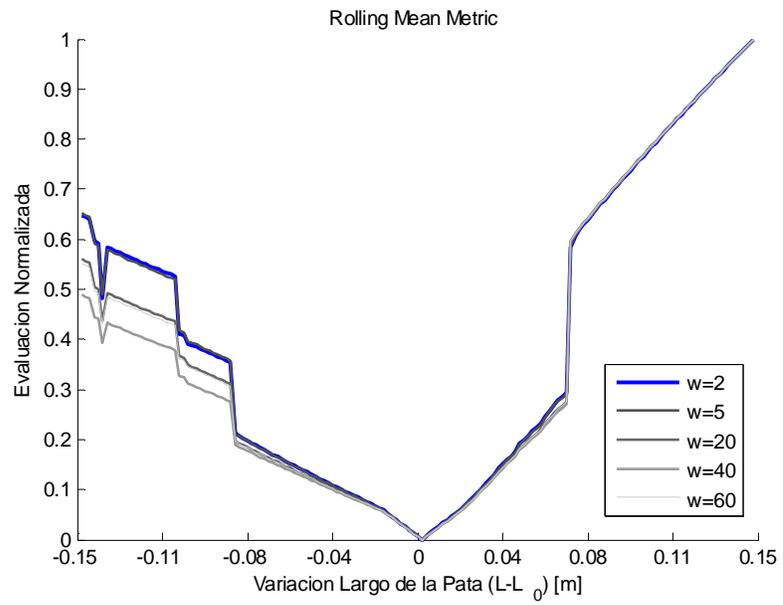


Figura 22. Patrón de evaluación para distintos valores de  $w$  con  $h=80$ .

#### *4.2.5 Nueva métrica propuesta*

Las métricas antes mencionadas (Euclidiana y Métrica de Promedio Rodante) tienen algunas limitaciones en cuanto a su robustez, por un lado la norma euclidiana tiende a fallar cuando los sistemas estudiados son muy complejos y se produce la divergencia de las señales, ver figura 5.

La métrica de promedio rodante, ofrece una solución al problema de la divergencia de las señales, considerando solo una pequeña ventana al inicio de la simulación. Sin embargo este método produce una gran sensibilidad a pequeñas diferencias en la posición inicial entre el robot real y el modelo candidato.

A su vez ambas métricas fallan cuando se produce un desfase entre la señal del candidato y la del robot real, ya que se pierde cualquier correlación directa entre las señales.

Debido a estas limitaciones se plantea la necesidad de contar con una métrica de naturaleza estadística que sea robusta al enfrentar problemas como la divergencia de señales, diferencias en la posición inicial y desfase entre las señales.

#### *4.2.6 Discriminante de Kullback*

En una aproximación estadística al problema de identificación de modelos se considera que los datos recolectados por un sensor son solo un resultado posible de una variable aleatoria que sigue cierta distribución. Esto implica que al comparar dos señales no se debe comparar la señal misma (que es solo un resultado posible de una variable aleatoria), si no que se debe comparar la distribución que siguen los valores entregados por el sensor.

El análisis de la distribución que siguen los valores entregados por cada sensor, en lugar de la señal misma, otorga beneficios en cuanto a robustez, considerando los problemas planteados anteriormente.

En cuanto a la comparación de distribuciones (o histogramas en su versión discreta) se propone el uso del Discriminante de Kullback. Este instrumento de comparación ha sido ampliamente utilizado para el problema de identificación de modelos y en teoría de la información, ver sección 2.5.

Este Discriminante entrega una medida de la probabilidad de que dos histogramas obedezcan al mismo modelo. Esta medida de probabilidad puede utilizarse como una medida de cuan parecidos son dos robots solo estudiando la distribución que siguen las señales entregadas por sus sensores.

La formulación discreta del Discriminante de Kullback es:

$$D_{KL}(P//Q) = \sum_i P(i) \log \left( \frac{P(i)}{Q(i)} \right) \quad (11)$$

Donde P y Q corresponden a distribuciones de probabilidad de una variable aleatoria discreta.

#### 4.2.7 Detalle experimento N° 2

Con el objetivo de estudiar la monotonía del patrón de la evaluación obtenida utilizando la nueva métrica propuesta, se repitió el Experimento N° 1, ahora incluyendo la evaluación mediante el Discriminante de Kullback.

En la figura 25 se muestra el patrón de evaluación obtenido con el Discriminante de Kullback, se observa que esta métrica entrega valor mínimo para el modelo real (condición indispensable para una buena métrica), además si bien exhibe cierta monotonía, esta no es mejor que la entregada por las otras dos métricas, ver figuras 23 y 24.

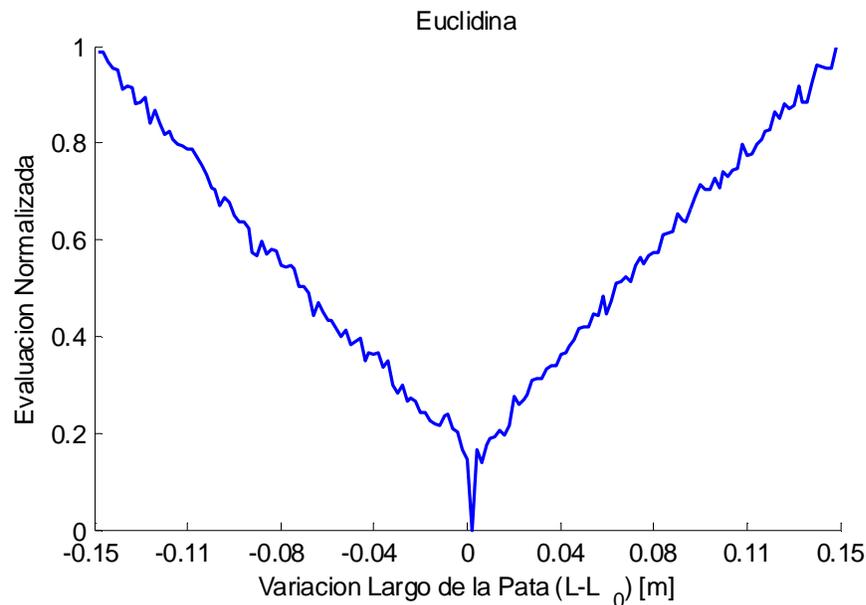


Figura 23. Patrón de evaluación utilizando métrica Euclidiana.

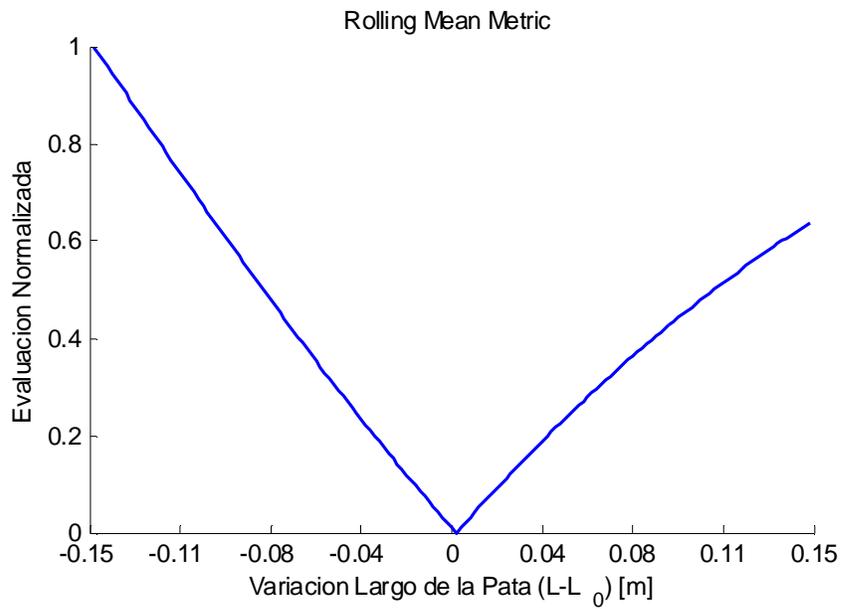


Figura 24. Patrón de evaluación utilizando Rolling Mean Metric con parámetros  $h=5$  y  $w=2$ .

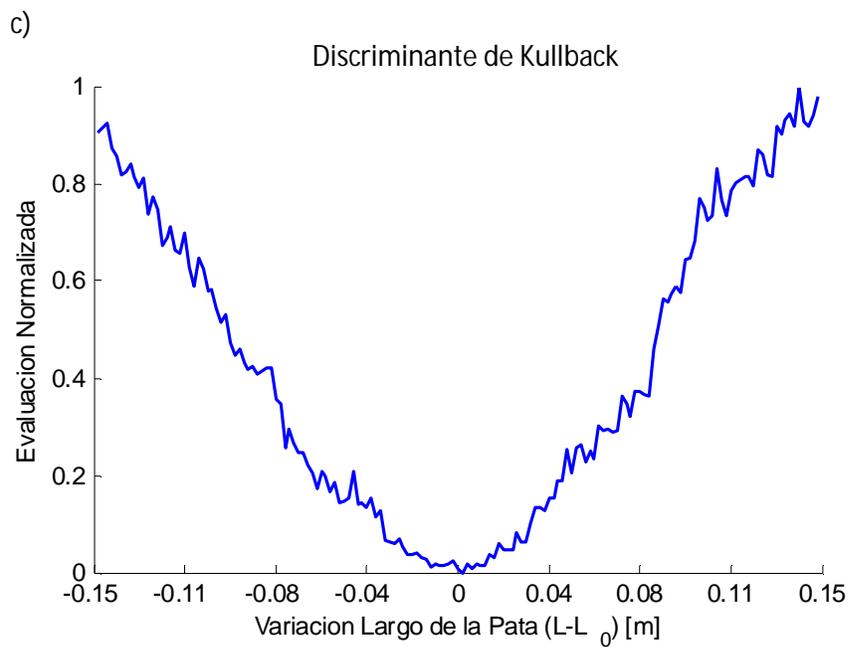


Figura 25. Patrón de evaluación utilizando Discriminante de Kullback.

### 4.2.8 Detalle experimento N° 3

Con el objeto de estudiar las posibles mejoras que podría presentar la nueva métrica en cuanto a robustez a la hora de evaluar señales que presentan desfase temporal o variaciones en la posición inicial del robot, se plantea el siguiente experimento.

En esta experiencia se utiliza el mismo simulador del robot trepen del experimento N° 1, es decir el mismo robot real y la misma población de candidatos, sin embargo se realiza un cambio a la hora de evaluar las señales obtenidas, en este caso se aplicara la métrica a un sub intervalo de la señal asociada a cada robot.

Para una señal de largo  $L$ , se utiliza un sub intervalo de largo  $D$ , escogiendo al azar los valores  $k_r$  y  $k_c$  dentro del intervalo  $[D/2, L-D/2]$ . Así los intervalos para el robot real y para los robots candidatos quedan definidos de la siguiente manera. Ver figura 26:

$$\begin{aligned} \text{Intervalo 1} &= [k_r - D/2, k_r + D/2] \\ \text{Intervalo 2} &= [k_c - D/2, k_c + D/2] \end{aligned} \quad (13)$$

Este procedimiento busca acercarse al caso de que el sensor del robot real se enciende en un momento y durante un tiempo arbitrarios, y lo mismo con los robots candidatos. Entonces en este caso se introduce un desfase de las señales a evaluar (real y candidatos).

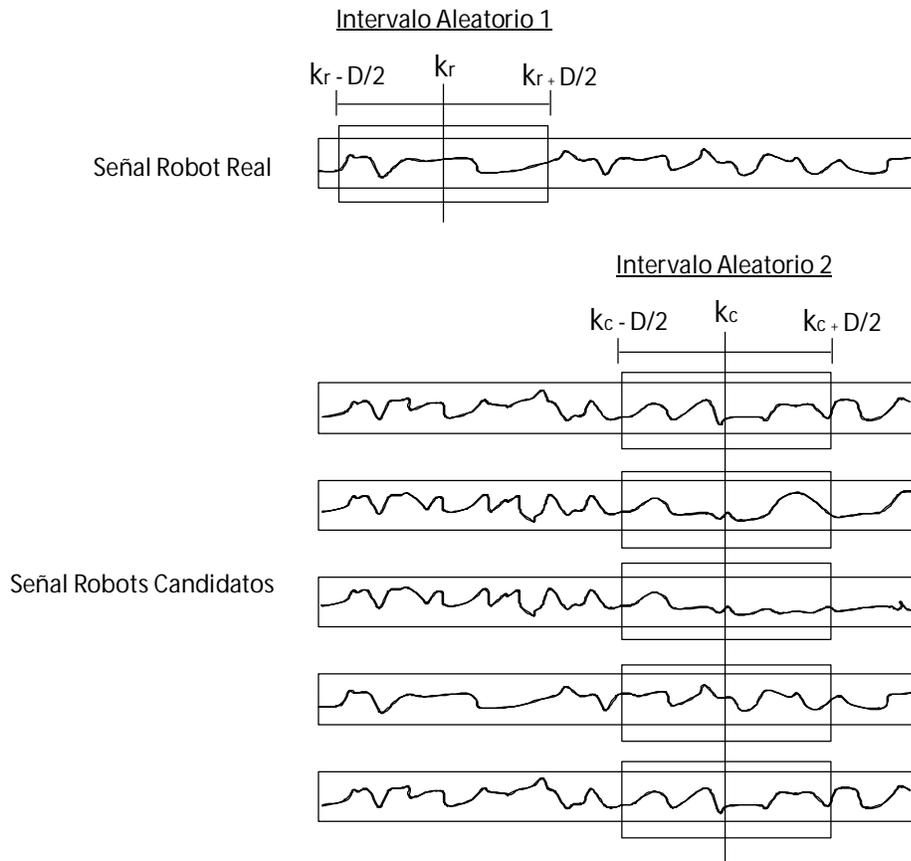
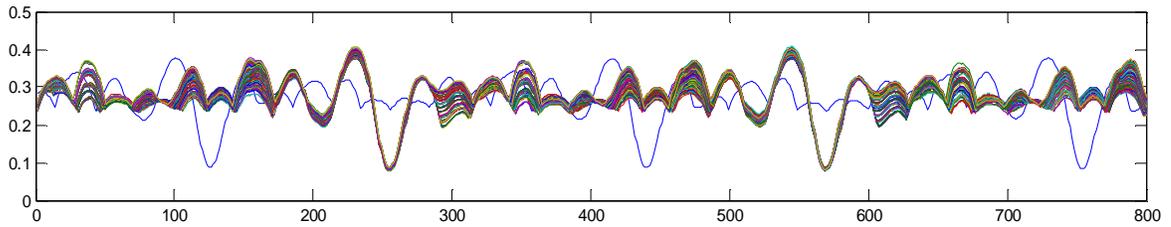


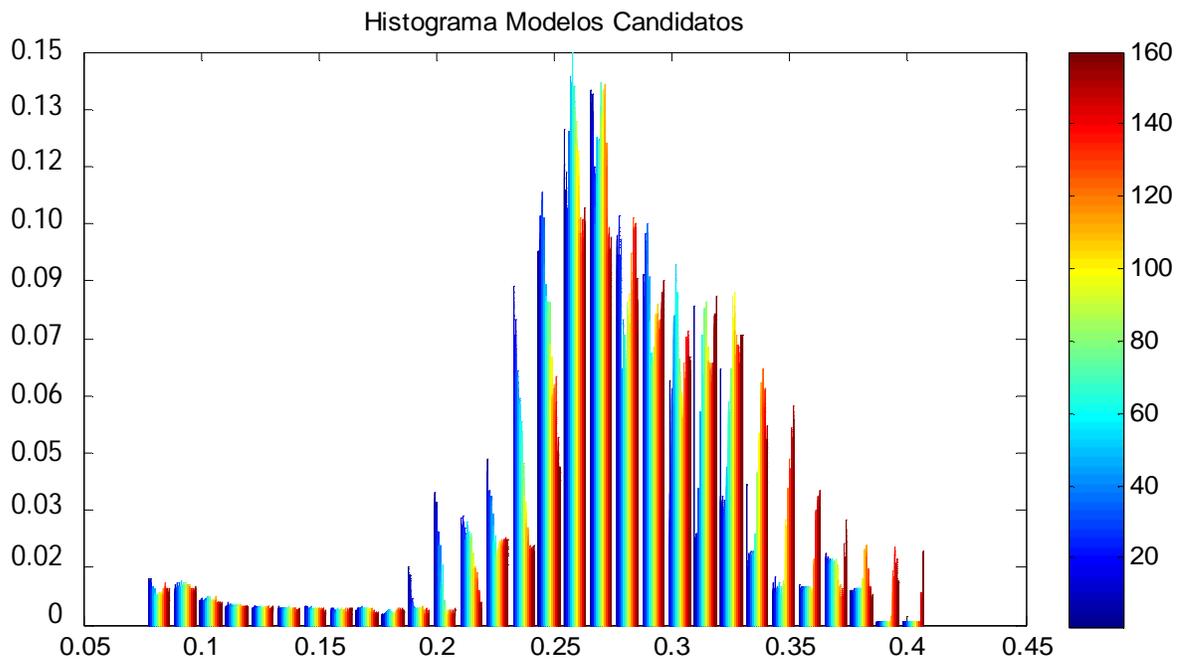
Figura 26. Esquema representativo del análisis de señales en sub intervalos aleatorios.

La evaluación de cada métrica en intervalos aleatorios, se repite aleatoriamente 12 veces obteniéndose los patrones de evaluación para cada métrica, ver figuras 29, 30, 31. La figura 27 muestra las series de tiempo obtenidas para un par de intervalos en particular, la señal azul corresponde al robot real.



**Figura 27. Series de tiempo obtenidas para un par de intervalos particular (uno de los 12 pares de intervalos).**

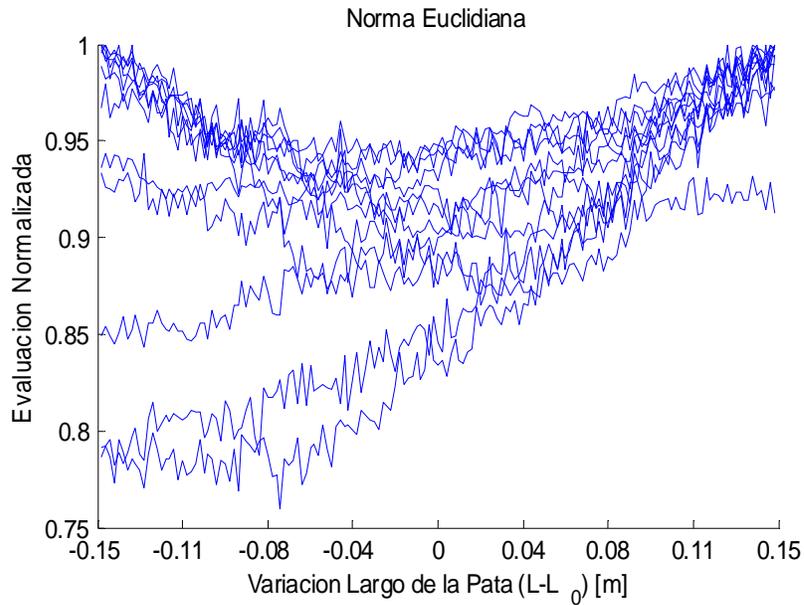
Los histogramas se construyen con 30 bins y se normalizan antes de la aplicación de la métrica. La figura 28 muestra los histogramas empleados para la comparación con Discriminante de Kullback. Cada color representa el histograma asociado a un robot en particular (160 en total).



**Figura 28. Histogramas de los 160 modelos candidatos, cada color representa un modelo distinto.**

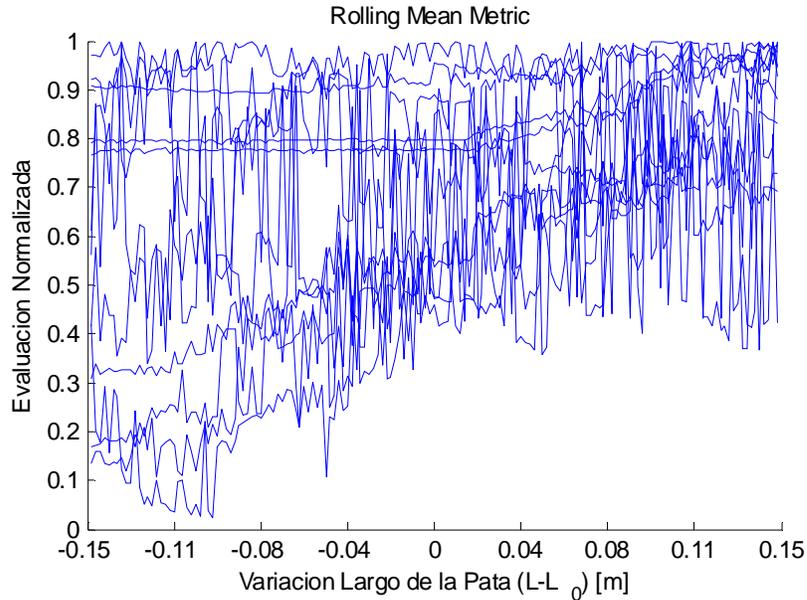
En la figura 29 se muestran los patrones de evaluación obtenidos utilizando la métrica Euclidiana. Cada curva corresponde a un par específico de intervalos en los que se evalúa la señal (un intervalo para la señal del robot real y un intervalo para la señal de los modelos candidatos).

Se observa que ninguna de las curvas presenta un mínimo en cero, es decir la métrica Euclidiana no es capaz de detectar que un modelo es igual al robot real.

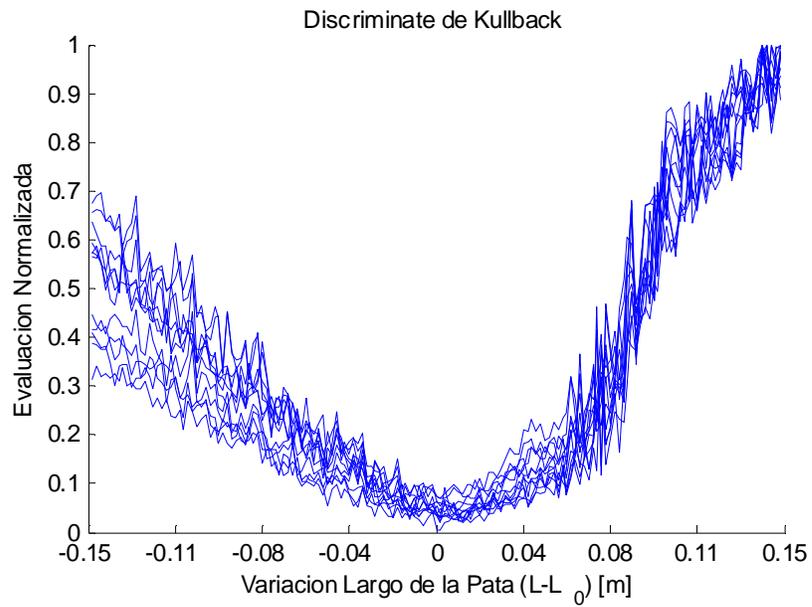


**Figura 29. Patrones de evaluación obtenidos mediante la métrica Euclidiana.**

La figura 30 muestra los patrones de evaluación obtenidos a través de la Rolling Mean Metric. Se observa que esta métrica tampoco entrega patrones con un mínimo en cero, es decir no es capaz de detectar cuando un modelo candidato es igual al robot real. Además prácticamente no se observa monotonía en las curvas obtenidas.



**Figura 30. Patrones de evaluación obtenidos mediante la Rolling Mean Metric.**



**Figura 31. Patrones de evaluación obtenidos mediante Discriminante de Kullback.**

El patrón de evaluación obtenido utilizando el Discriminante de Kullback, figura 31, muestra claramente que las curvas tienden a presentar un mínimo en torno al cero, es decir para todos los sub intervalos estudiados esta métrica logra detectar aproximadamente cual es el modelo candidato más parecido al robot real.

### **4.3 Algoritmo de auto-modelado incorporando la nueva métrica**

El objetivo de este experimento es comprobar si las características de flexibilidad y robustez mostradas por el Discriminante de Kullback, tienen un efecto positivo a la hora de realizar una búsqueda de modelos a través de algoritmos genéticos.

#### **4.3.1 Detalle experimento N° 3**

Simulador:

El entorno de simulación se mantiene invariante con respecto al experimento N° 1, el tiempo de simulación para cada candidato es de 60 [s] equivalente a series de tiempo con 3000 puntos.

El sub intervalo donde se aplica la métrica se fija en 16 [s] equivalente a 800 puntos en la serie de tiempo.

Búsqueda con algoritmos genéticos:

La búsqueda con algoritmos genéticos comienza con una población inicial de 150 modelos candidatos que se seleccionan a lo largo de 15 generaciones.

El espacio de búsqueda en este caso corresponde al largo de una pata del robot trepen, osea el genoma utilizado en este caso corresponde a un valor real  $\{L\}$ , perteneciente al intervalo  $[0.05m, 0.35m]$ . El largo de la pata real se fija en  $L_0 = 0.2$ .

Para la función objetivo se utilizó el Discriminante de Kullback, la métrica Euclidiana y la Rolling Mean Metric. Utilizando los intervalos aleatorios descritos en el experimento N° 3, figura 26.

La figura 32 muestra las evaluaciones realizadas por el algoritmo de búsqueda en tres experimentos distintos utilizando la norma Euclidiana, cada uno asociado a un par de sub intervalos aleatorios. Se observa que la métrica no entrega buenos patrones de evaluación y con mínimos alejados del cero, lo que significa que la métrica no está siendo efectiva para detectar al modelo más parecido al robot real. El resultado de la búsqueda en este caso se muestra en la tabla 3. El Error promedio de los resultados con respecto al robot real es de 48%.

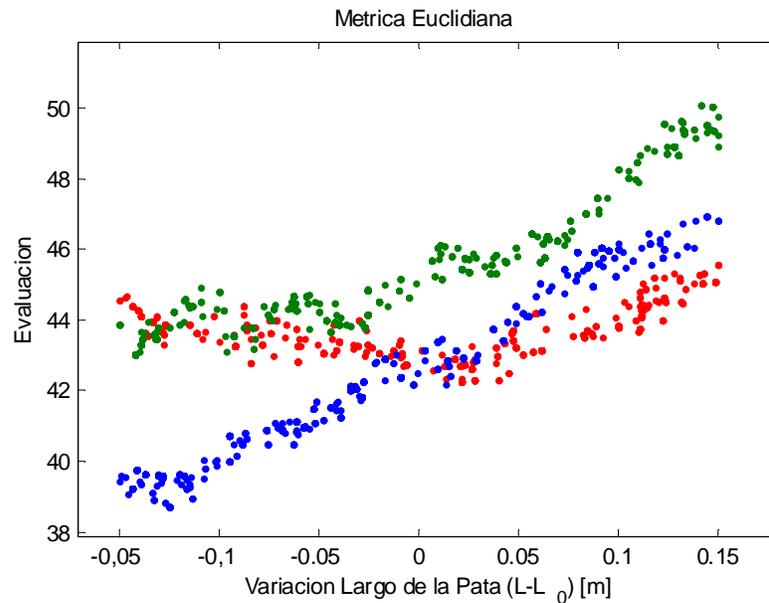
La figura 33 muestra los resultados obtenidos utilizando Rolling Mean Metric, en este caso tampoco se observa mínimos en cero para ninguna de las tres series estudiadas. El resultado de la búsqueda, ver tabla 4, muestra un error promedio de 60.1 %

La búsqueda realizada utilizando el Discriminante de Kullback, figura 34, se observa claramente que las tres series tienden a presentar mínimos cerca del cero, osea para los tres pares de sub intervalos aleatorios utilizados, esta métrica logra detectar aproximadamente cual es el modelo más parecido al robot real. El resultado de la búsqueda en este caso se muestra en la tabla 5, exhibiendo un error promedio de 5.3%.

Los valores obtenidos finalizado el algoritmo de búsqueda utilizando métrica Euclidiana son:

Serie 1: $L_1 = 0.2215 \text{ m}$
Serie 2: $L_2 = 0.0577 \text{ m}$
Serie 3: $L_3 = 0.0745 \text{ m}$

**Tabla 3. Resultados del parámetro de la búsqueda de las 3 series estudiadas utilizando métrica Euclidiana.**

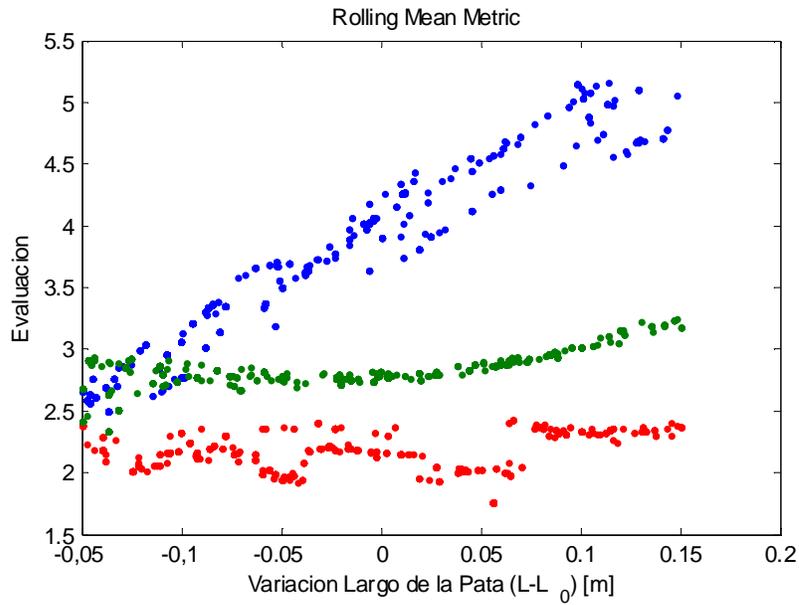


**Figura 32. Patrón de evaluaciones del algoritmo de búsqueda utilizando métrica Euclidiana, Serie1: rojo, Serie2: verde, Serie 3: azul.**

Los resultados obtenidos con Rolling Mean Metric son:

Serie 1: $L_1 = 0.2563 \text{ m}$
Serie 2: $L_2 = 0.0329 \text{ m}$
Serie 3: $L_3 = 0.0630 \text{ m}$

**Tabla 4. Resultados del parámetro de búsqueda de las 3 series estudiadas utilizando Rolling Mean Metric.**

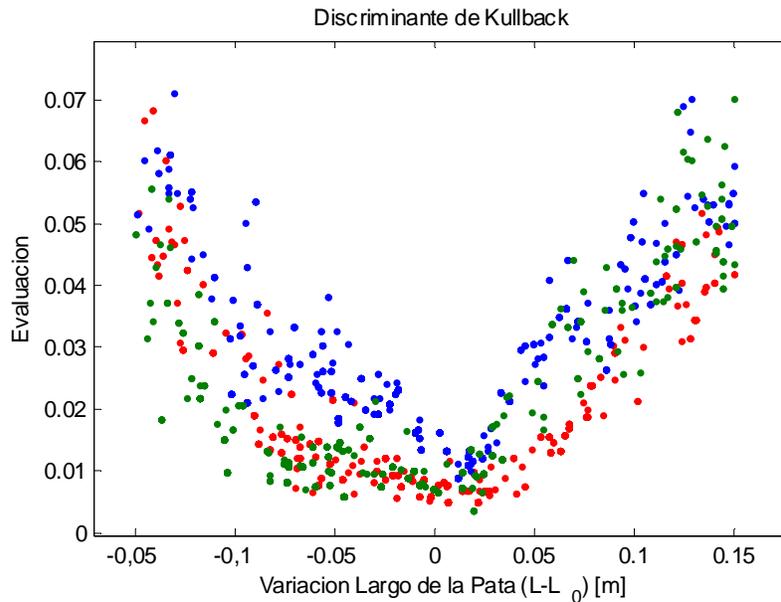


**Figura 33. Patrón de evaluaciones del algoritmo de búsqueda utilizando Rolling Mean Metric. Serie 1: rojo, Serie 2: verde, Serie 3: azul.**

Los valores obtenidos por el algoritmo de búsqueda utilizando Discriminante de Kullback para las tres series son:

Serie 1: $L_1 = 0.2060 \text{ m}$
Serie 2: $L_2 = 0.1850 \text{ m}$
Serie 3: $L_3 = 0.2109 \text{ m}$

**Tabla 5. Resultados de la búsqueda utilizando Discriminante de Kullback.**



**Figura 34. Patrón de evaluaciones del algoritmo de búsqueda utilizando Discriminante de Kullback. Serie 1: rojo, Serie 2: verde, Serie 3: azul.**

La figura 35 muestra que la evaluación promedio a través de las generaciones disminuye mucho más rápido en el caso del Discriminante de Kullback, en particular luego de 15 generaciones la evaluación promedio disminuye en un 65%. Una tendencia similar se observa para la varianza de la población, ver figura 36.

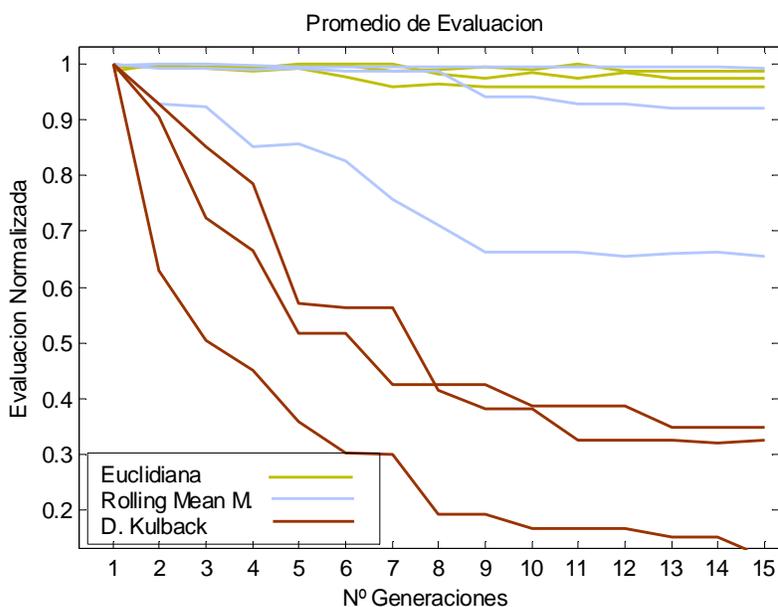


Figura 35. Evolución de la evaluación promedio a través de las generaciones para las tres series estudiadas.

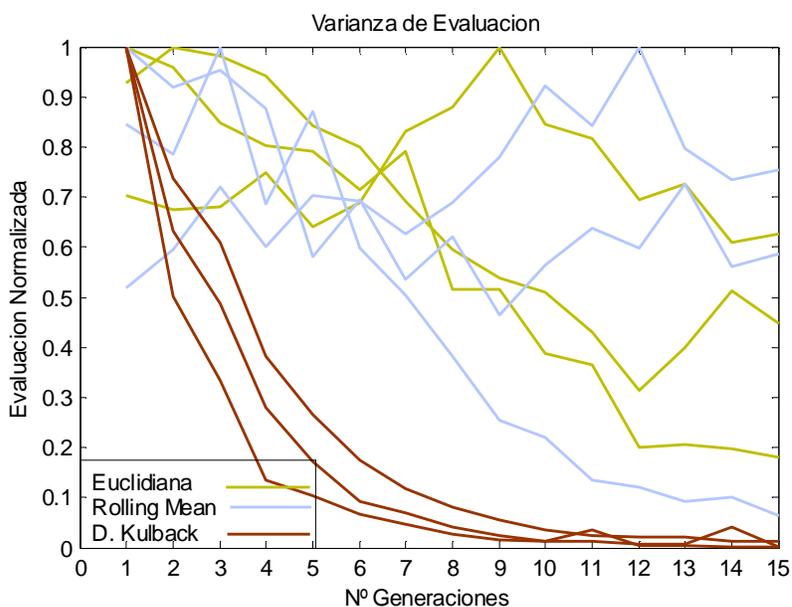


Figura 36. Evolución de la varianza a través de las generaciones para las tres series estudiadas.

En cuanto a la evolución del valor promedio parámetro de búsqueda a través de las generaciones, en la figura 37 se muestra que en el caso del Discriminante de Kullback las tres series presentan convergencia cerca del valor  $r_{real} = 0.2 m$ , lo que es esperable para una buena métrica. Para el caso de las otras dos métricas, éstas convergen a valores más alejados de  $r_{real} = 0.2 m$ .

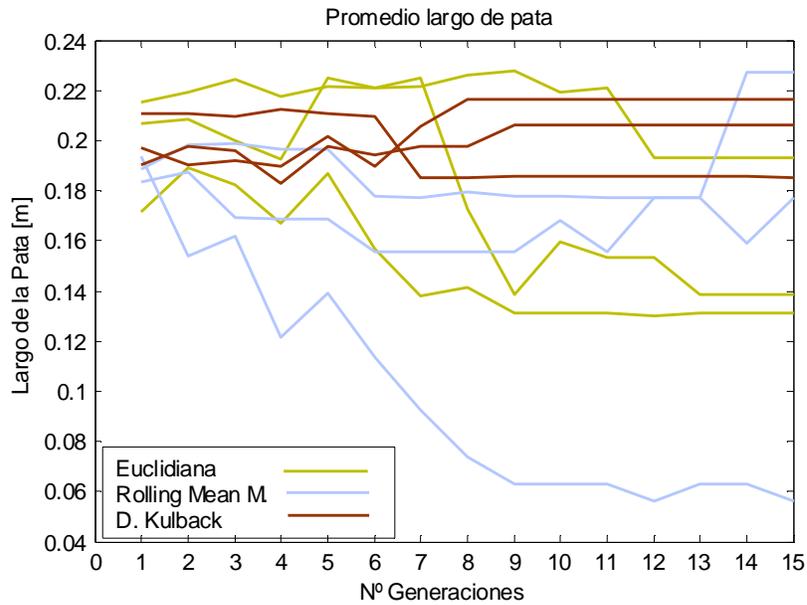


Figura 37. Evolución parámetros de búsqueda a través de las generaciones para las tres series estudiadas.

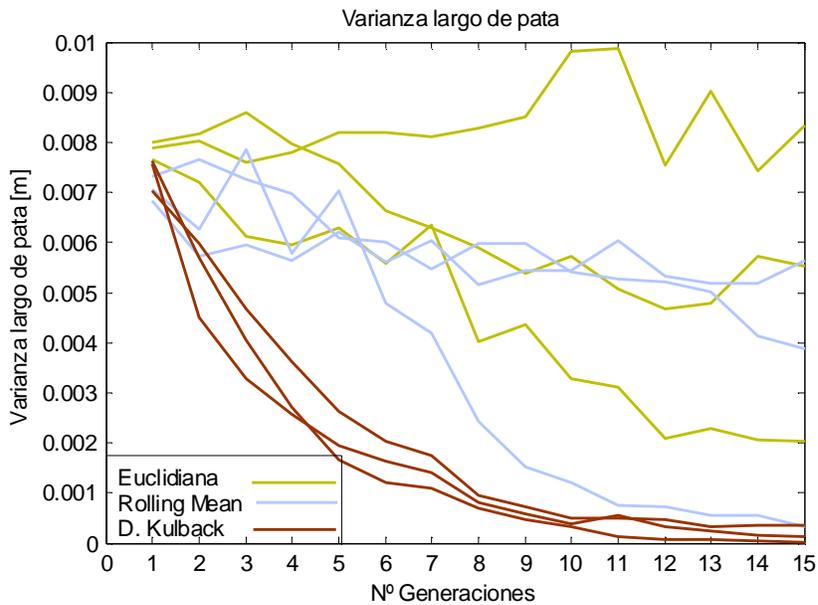


Figura 38. Evolución de la varianza de la población a través de las generaciones para las tres series estudiadas.

En cuanto a la varianza de la población, la figura 38 muestra que una convergencia más rápida para el Discriminante de Kullback que para las otras dos métricas. En particular se observa que luego de la octava generación la varianza permanece más o menos constante, este fenómeno se condice con la evolución

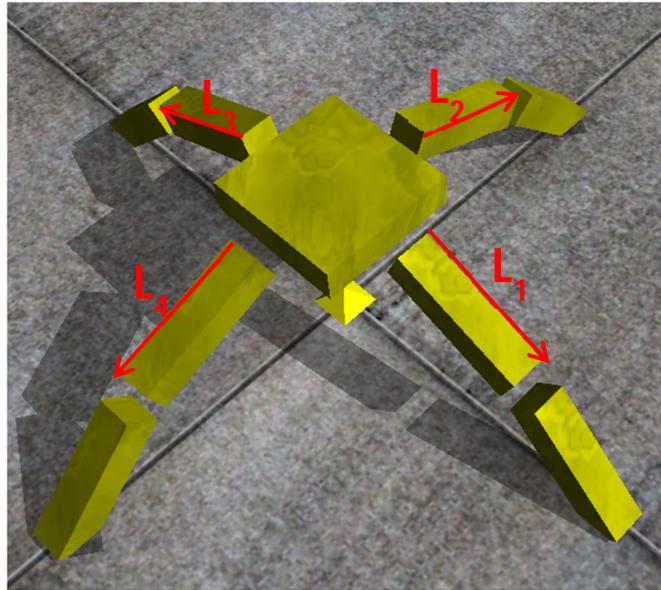
promedio del parámetro de búsqueda, ver figura 37, donde se observa también una estabilización del parámetro de búsqueda luego de la octava generación.

### 4.3.2 Detalle experimento N° 4

El objetivo de este experimento es estudiar el comportamiento de la nueva métrica utilizando espacios de búsqueda más amplios.

El espacio de búsqueda se define como los largos de cuatro extremidades del robot trepen, osea el genoma utilizado corresponde a 4 valores reales  $\{L_1, L_2, L_3, L_4\}$ , que pertenecen al intervalo  $[0.05, 0.35]$ . Ver figura 39.

Las características del simulador y la búsqueda permanecen invariantes con respecto al experimento N° 3. Solo varía en cuanto al tamaño de la población, que queda de 200 individuos y el número de generaciones permanece fija en 35. En cuanto a las señales estudiadas, el largo total de las señales se fija en de 2000 puntos mientras que el largo de los sub intervalos se fija en 1300 puntos.



**Figura 39. Descripción de los parámetros de búsqueda, corresponde al largo de cuatro extremidades del robot Trepen.**

El patrón de evaluaciones del algoritmo de búsqueda se presenta en la figura 40, en este caso la abscisa corresponde a la distancia euclidiana entre el vector de parámetros que representa a un candidato y el vector de parámetros que representa al robot real. La ordenada corresponde a la evaluación normalizada entregada por la métrica en cada evaluación del algoritmo. La siguiente expresión define la distancia euclidiana utilizada en este caso.

$$D = (L_{1,ca} - L_{1,re})^2 + (L_{2,ca} - L_{2,re})^2 + (L_{3,ca} - r_{3,re})^2 + (L_{4,ca} - L_{4,re})^2 \quad (12)$$

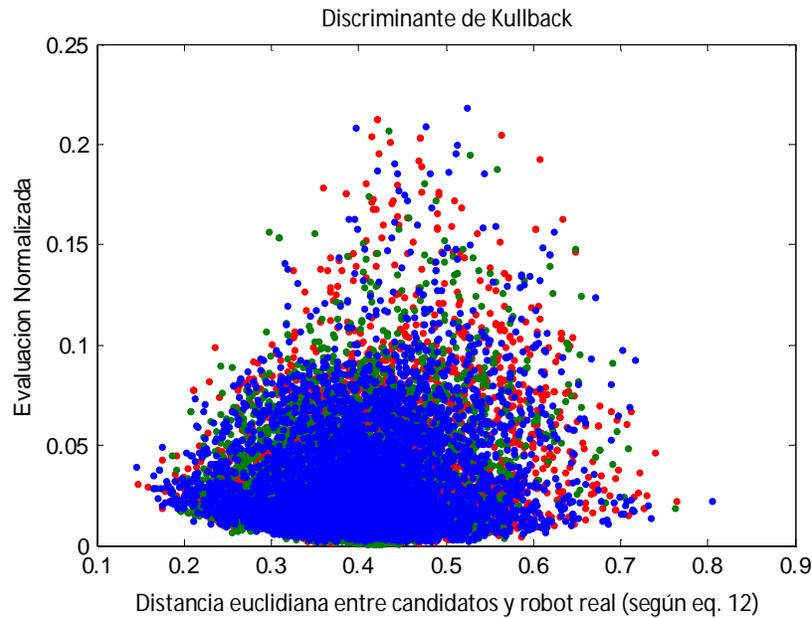
Resultado obtenido luego de finalizado el algoritmo de búsqueda es:

Serie 1	Serie 2	Serie 3
$L_1 = 0.1908 \text{ m}$	$L_1 = 0.1901 \text{ m}$	$L_1 = 0.2202 \text{ m}$
$L_2 = 0.2035 \text{ m}$	$L_2 = 0.1984 \text{ m}$	$L_2 = 0.2417 \text{ m}$
$L_3 = 0.1945 \text{ m}$	$L_3 = 0.2108 \text{ m}$	$L_3 = 0.2097 \text{ m}$
$L_4 = 0.2136 \text{ m}$	$L_4 = 0.2203 \text{ m}$	$L_4 = 0.2375 \text{ m}$

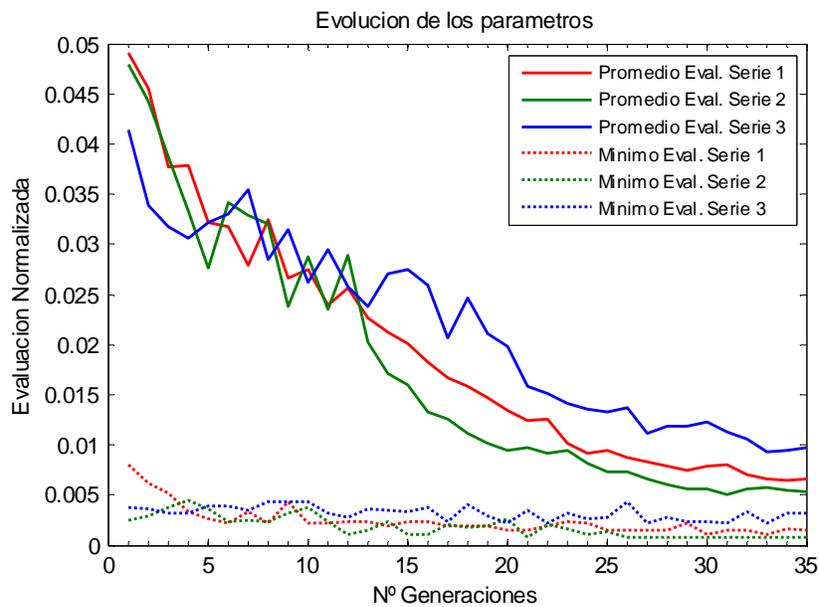
**Tabla 6. Resultados búsqueda de 4 parámetros utilizando Discriminante de Kullback**

La figura 40 muestra una relación positiva entre la evaluación de la métrica y el valor  $D$ . Este es un resultado esperable para una buena métrica ya que significa que se está reflejando de buena manera cuan distinto es un robot candidato del robot real.

En cuanto al resultado de la búsqueda, ver tabla 6, se observa que el error promedio de los parámetros con respecto al robot real es de 4%, 5.3% y 13.6% para las serie 1, 2 y 3 respectivamente. Esto equivale a un error promedio total de 7.6 %.

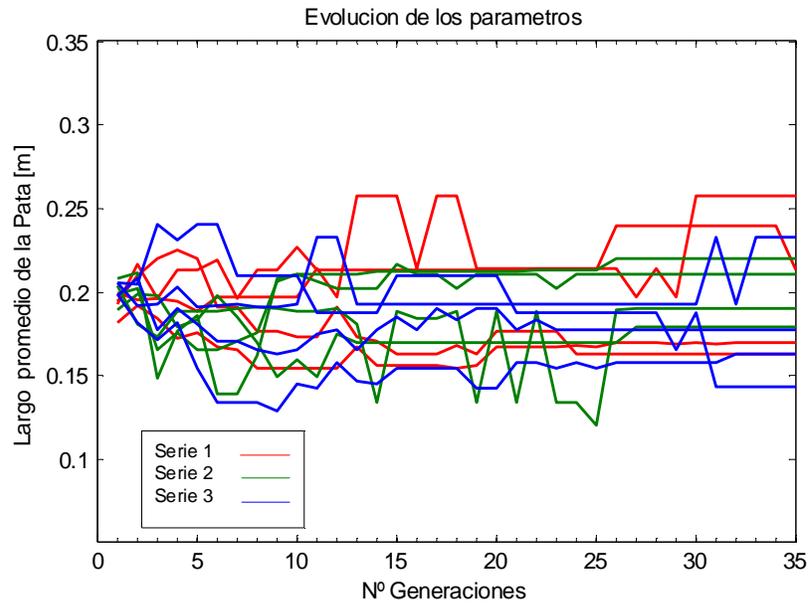


**Figura 40. Patrón de evaluaciones del algoritmo de búsqueda utilizando Discriminante de Kullback.**



**Figura 41. Evolución de las evaluaciones a través de las generaciones. Utilizando métrica euclidiana.**

En cuanto a la convergencia de la evaluación promedio a través de las generaciones, la figura 41 muestra un buen patrón de convergencia para las tres series estudiadas, en particular luego de 35 generaciones la evaluación promedio disminuye en un 80%. Sin embargo esta tendencia es menos pronunciada para el mínimo evaluado (figura 41, línea punteada) que para la evaluación promedio a través de las generaciones.



**Figura 42. Evolución del radio promedio de los puntos de control a través de las generaciones.**

La figura 42 se observa la evolución de los parámetros de búsqueda promedio a través de las generaciones, se muestra que las tres series exhiben grandes oscilaciones alrededor del valor  $L = 0.2 m$ . Osea no se presenta una convergencia tan buena como el caso de la búsqueda de un solo parámetro, ver figura 37.

## 4.4 Nueva representación genérica de robots

### Objetivo de una nueva representación

Con el objeto de aumentar la flexibilidad y campo de acción del algoritmo de auto modelado se plantea una nueva forma de representar la geometría de partes individuales de un robot.

Esta representación debe satisfacer algunos requisitos:

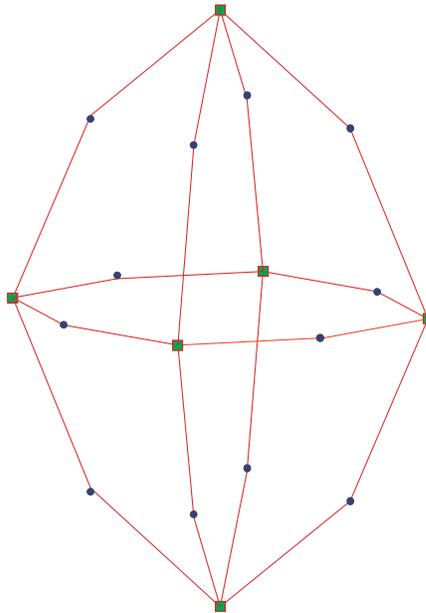
**Economía:** Se deben utilizar pocos parámetros para caracterizar a la pieza, esto con el propósito de que el espacio de búsqueda de modelos no crezca demasiado.

**Formas complejas:** Debe ser capaz de describir curvas suaves con precisión y formas amorfas en general.

### Representación Propuesta

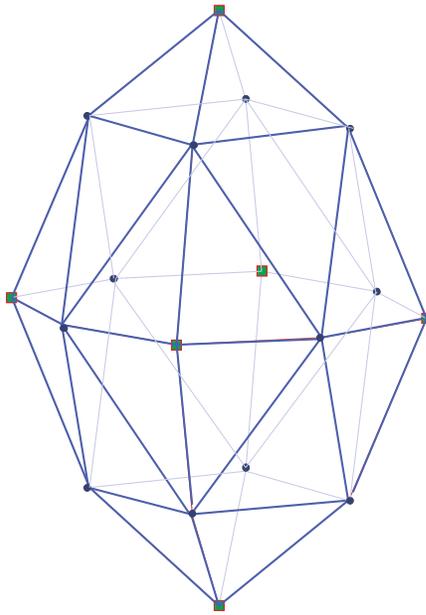
La representación propuesta considera una parametrización de la forma del objeto a través de 6 puntos de control que forman un prisma como el que se observa en la figura 43. Cada punto de control se describe a través de 3 números reales que corresponden a sus coordenadas  $x,y,z$ .

Los 6 puntos de control son interpolados por curvas spline cúbicas, (con un parámetro de tensión  $t=0$ ) que generan puntos intermedios. Los puntos de control (puntos verdes) y los puntos interpolados (puntos azules) se muestran en la figura 43.



**Figura 43. Descripción de la nueva representación propuesta. Se muestran los 6 puntos de control en verde, las curvas de interpolación y el mallado de triángulos.**

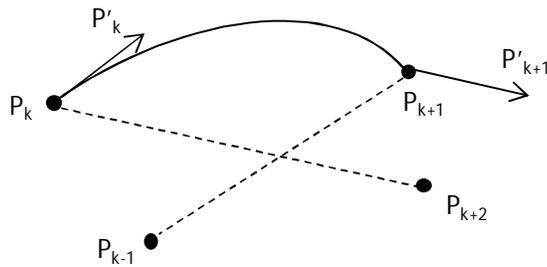
Posteriormente se genera una malla de triángulos utilizando tanto los puntos de control como los puntos interpolados, ver figura 44. La topología de la malla de triángulos permanece constante para cualquier posición de los puntos de control.



**Figura 44. Descripción del malla de triángulos de la nueva representación.**

En total esta representación requiere de 18 parámetros, sin embargo para ampliar el rango de formas posibles se podría incluir un parámetro de tensión para la interpolación entre cada punto.

La interpolación entre los puntos de control se realiza utilizando splines cúbicos cardinales de grado 3. La interpolación cardinal se caracteriza porque la curva de interpolación depende de los puntos de control y de un parámetro de tensión. Este parámetro de tensión se fijó en cero para todos los experimento realizados.

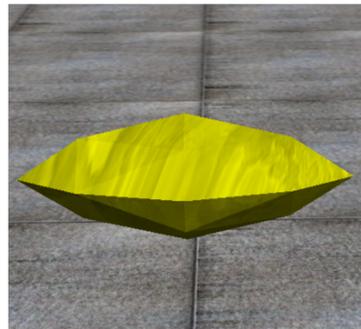


**Figura 45. Esquema descriptivo de splines Cardinales para serie de cuatro puntos.**

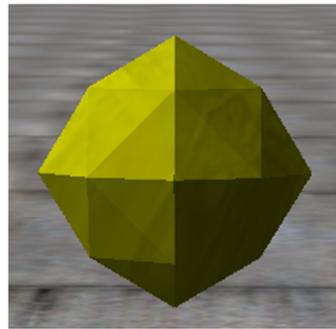
$$P(x) = [x^3 x^2 x 1] \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} P(0) \\ P(1) \\ \frac{1-t}{2} (P_{k+1} - P_{k-1}) \\ \frac{1-t}{2} (P_{k+2} - P_k) \end{bmatrix} \quad (14)$$

#### 4.4.1 Algunas representaciones posibles

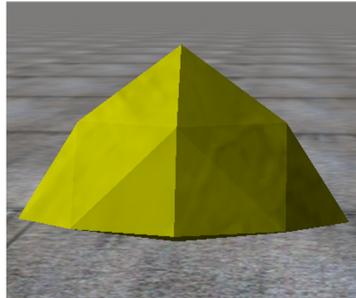
En la figura 46 se muestran algunas de las formas que se pueden conseguir con la parametrización propuesta.



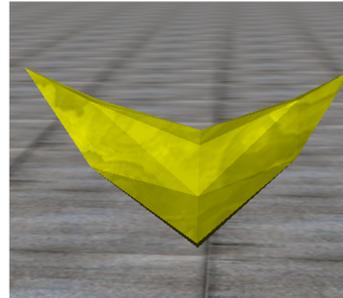
Forma N°1



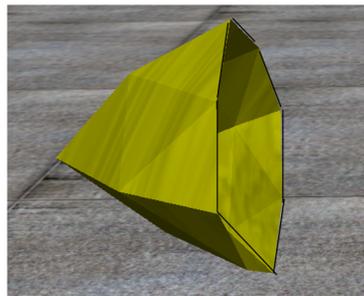
Forma N° 2



Forma N° 3



Forma N° 4



Forma N° 5

Figura 46. Formas que se pueden caracterizar utilizando la representación con 6 puntos de control.

#### 4.4.2 Robot que incorpora la nueva representación

A continuación se describen las características del robot diseñado para el estudio de la nueva representación. El diseño de este robot está orientado a destacar lo mejor posible las propiedades geométricas de un objeto amorfo.

Se trata de un triciclo cuya rueda delantera es reemplazada por un sólido amorfo descrito con la representación basada en spines, figura 47. Las dos ruedas traseras son cilindros y el cuerpo central es un paralelepípedo. Las dos ruedas traseras y la rueda delantera están fijas al cuerpo central mediante articulaciones de bisagra.



**Figura 47. Robot triciclo que incorpora nueva representación propuesta.**

Este robot dispone de un sensor de altura ubicado en el centro de masa del cuerpo central del robot. En cuanto al controlador este corresponde a un motor ubicado en el eje delantero que impone una velocidad angular constante a la rueda de 3 [rad/s].

En cuanto al sólido amorfo, éste es controlado por los 6 puntos de control descritos en la figura 43.

Las características del entorno de simulación son las mismas que las descritas en el experimento N° 1.

## 4.5 Algoritmo de auto modelado incorporando nueva representación

En esta etapa se implementó un algoritmo de auto modelado incorporando la nueva métrica propuesta, descrita en la sección 5.2.

### 4.5.1 Detalle experimento N° 5

El objetivo de este experimento se estudiar el comportamiento de un algoritmo de búsqueda utilizando un robot que incorpore la nueva representación propuesta. El espacio de búsqueda, en este caso son las coordenadas de los puntos de control del solido amorfo (superficie de interpolación).

Simulador:

El robot utilizado en esta simulación un triciclo cuya rueda principal ha sido reemplazada por un salido amorfo, figura 47.

La simulación para cada candidato dura 5 [s] produciendo series de tiempo de 250 puntos. Las características físicas del entorno de simulación son las mismas que las utilizadas en experimentos anteriores, experimento N° 1.

El parámetro variable en este experimento corresponde al radio de uno de los puntos de control del solido amorfo (el ángulo permanece constante), como se muestra en la figura 48.

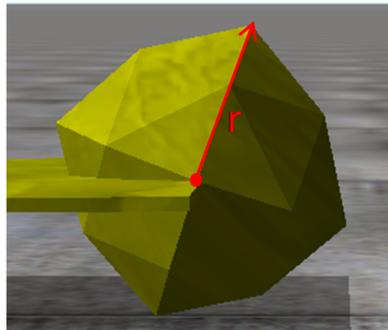


Figura 48. Parámetro variable durante el algoritmo de búsqueda

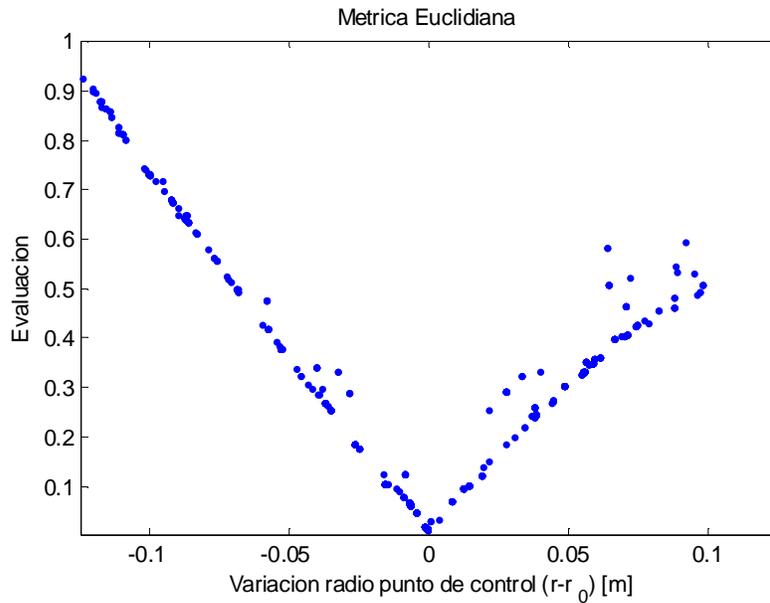
Búsqueda con algoritmos genéticos:

La búsqueda con algoritmos genéticos comienza con una población inicial de 150 modelos candidatos que se seleccionan a lo largo de 15 generaciones.

El espacio de búsqueda en este caso corresponde al radio de un punto de control del solido amorfo, ver figura 48. Osea el genoma utilizado en este caso corresponde a un valor real  $\{r\}$ , perteneciente al intervalo  $[0.02m, 0.25m]$ . El radio del punto de control en el robot real se fija en  $r_0 = 0.12$ .

La función objetivo utilizada en este caso consiste en una métrica euclidiana simple sobre el intervalo completo de las señales obtenidas.

La figura 49 muestra los puntos que se evalúan durante el procedimiento de búsqueda, y muestra una clara convergencia hacia el valor esperado;  $(r - r_0) = 0$ .



**Figura 49. Patrón de evaluaciones del algoritmo de búsqueda utilizando norma Euclidiana. Búsqueda de un parámetro del robot.**

La figura 49 muestra la evaluación de la métrica para los candidatos a lo largo del algoritmo de búsqueda. El valor obtenido finalmente es

$r = 0.1208 \text{ m}$
------------------------

**Tabla 7. Resultado búsqueda de 1 parámetro de la representación con splines.**

Como muestra la tabla 7 el resultado de la búsqueda es muy cercano al valor real, discrepa solo en un 0.7%.

En cuanto a la convergencia, la figura 50 muestra una clara disminución de la evaluación promedio a lo largo de las generaciones y muestra también una estabilización luego de la generación N° 11. Al llegar a esa generación se ha producido una disminución de la evaluación de un 95%.

En la figura 51 también se observa una estabilización del parámetro de búsqueda luego de la generación N° 11 y una convergencia del valor promedio del parámetro de búsqueda cerca del 0.12.

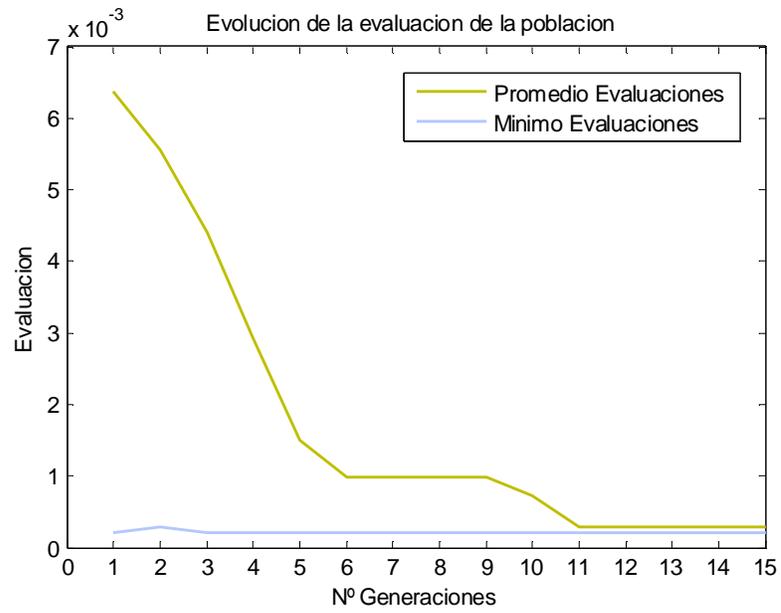


Figura 50. Evolución de las evaluaciones a lo largo del algoritmo de búsqueda utilizando norma Euclidiana.

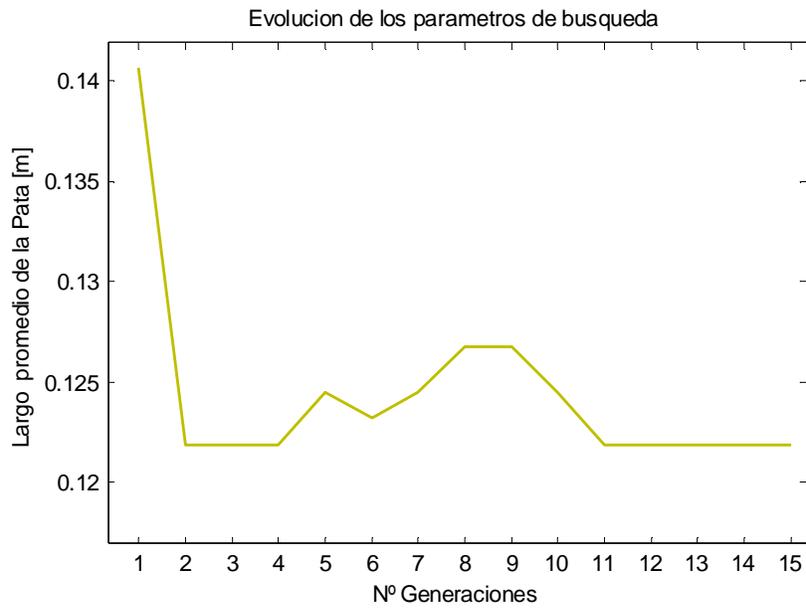


Figura 51. Evolución del radio promedio de del punto de control a través de las generaciones.

### 4.5.2 Detalle experimento N° 6

#### Simulador:

El robot utilizado en esta simulación un triciclo cuya rueda principal ha sido reemplazada por un solido amorfo, figura 47.

La simulación para cada candidato dura 5 [s] produciendo series de tiempo de 250 puntos. Las características físicas del entorno de simulación son las mismas que las utilizadas en experimentos anteriores, experimento N° 1.

El parámetro variable en este experimento corresponde a los radios de 4 puntos de control del solido amorfo (el ángulo permanece constante), como se muestra en la figura 52.

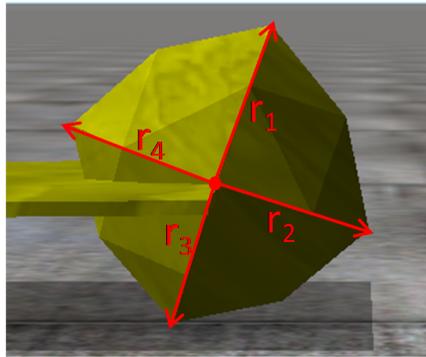


Figura 52. Descripción de los parámetros de búsqueda (foto rueda)

#### Búsqueda con algoritmos genéticos:

La búsqueda con algoritmos genéticos comienza con una población inicial de 600 modelos candidatos que se seleccionan a lo largo de 20 generaciones.

El espacio de búsqueda en este caso corresponde al radio de 4 puntos de control del solido amorfo, osea el genoma utilizado en este caso corresponde a 4 valores reales,  $r_{cand} = \{r_1, r_2, r_3, r_4\}$ , pertenecientes al intervalo [0.02m, 0.25m]. Los radios de los puntos de control en el robot real se fija en  $r_{real} = \{0.12, 0.12, 0.12, 0.12\}$ .

La figura 53 muestra el patrón de evaluación producido por el algoritmo de búsqueda, en este caso, para medir la eficiencia de la búsqueda la evaluación normalizada para cada modelo en función de la distancia euclidiana D, entre el vector que describe el modelo real y el vector que describe el modelo candidato.

$$D = (r_{1,ca} - r_{1,re})^2 + (r_{2,ca} - r_{2,re})^2 + (r_{3,ca} - r_{3,re})^2 + (r_{4,ca} - r_{4,re})^2 \quad (15)$$

Puede observarse también en la figura 53 que hay una relación positiva entre D y la evaluación entregada por la métrica, esto es un resultado esperable cuando ya que durante la búsqueda se está calificando de mejor manera a los modelos candidatos que más se asemejan al modelo real.

La figura 54 muestra la evolución de las evaluaciones a través de las generaciones, puede observarse que el promedio de evaluaciones decrece continuamente a lo largo del algoritmo de búsqueda disminuyendo en un 82% al llegar a la generación N° 19. El mínimo de las evaluaciones no muestra una disminución tan pronunciada como la del promedio.

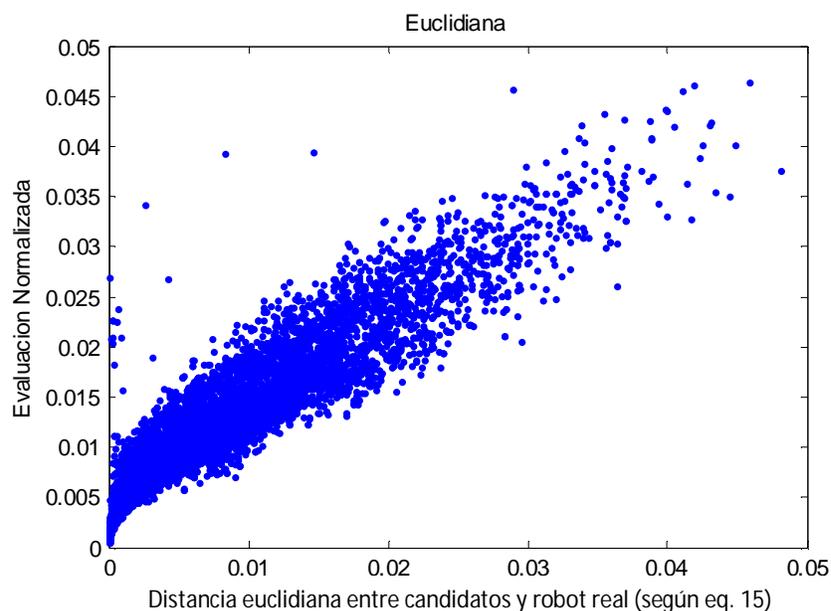
El resultado obtenido al final de la búsqueda es:

$r_1 = 0.1214 m$
$r_2 = 0.1208 m$
$r_3 = 0.1215 m$
$r_4 = 0.1186 m$

**Tabla 8. Resultado búsqueda de 4 parámetros de la representación con splines**

La figura 55 muestra la evolución de los parámetros de búsqueda a través de las generaciones, se observa que luego de la generación N° 13 el valor de los parámetros se estabiliza en valores cercanos al 0.12.

La tabla 8 muestra el resultado del algoritmo de búsqueda, se obtuvieron valores bastante parecidos al robot real, con un error promedio de 1%.



**Figura 53. Patrón de evaluaciones del algoritmo de búsqueda utilizando norma Euclidiana. Búsqueda de cuatro parámetros del robot.**

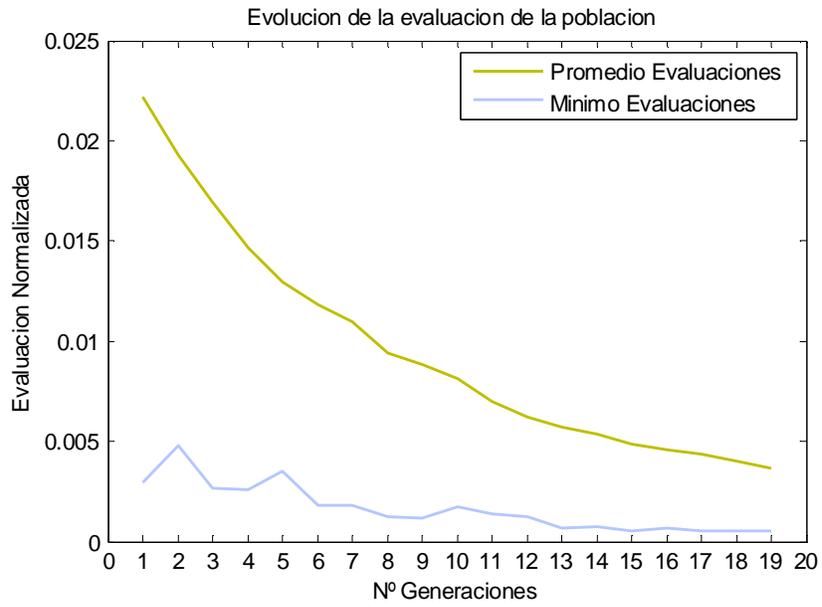


Figura 54. Evolución de las evaluaciones a través de las generaciones. Utilizando métrica euclidiana.

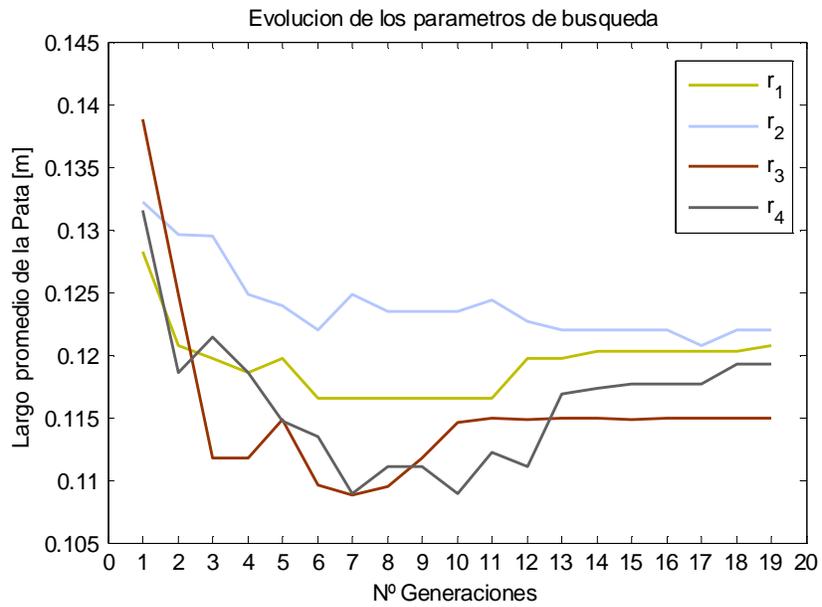


Figura 55. Evolución del radio promedio de los puntos de control a través de las generaciones.

## 5 Conclusiones

Se comprobó la funcionalidad del modelo de simulación para realizar experimentos y recolectar información útil de estos. Esta funcionalidad se utilizó para realizar un estudio de la eficacia de distintas métricas, nuevas representaciones y búsquedas con algoritmos genéticos.

El estudio de métricas existentes mostro que la métrica Euclidiana entrega patrones de evaluación con una monotonía aceptable, sin embargo la Rolling Mean Metric resulta en patrones de evaluación mucho más monótonos para valores de  $h$  pequeños. No se apreció un efecto importante del parámetro  $w$ .

El experimento N° 3 muestra claramente las debilidades de las métricas Euclidiana y Rolling Mean Metric en cuanto a robustez, ya que estas fallan al producirse un desfase de las señales. A su vez la nueva métrica propuesta muestra un muy buen desempeño al manejar señales desfasadas (evaluación en sub intervalos), ver figura 31.

Cuando se estudió el comportamiento de la nueva métrica en un algoritmo de búsqueda genética, se observa que esta produce una mejora considerable en la convergencia, ver figura 35 y figura 36. En cuanto a la precisión del resultado final de la búsqueda, El Discriminante de Kullback arroja un error promedio con respecto al robot real de 5.3% mientras que la métrica Euclidiana y Rolling Mean Metric arrojan errores promedio de 48% y 60.1% respectivamente.

En cuanto a la búsqueda utilizando 4 parámetros del robot trepen utilizando la nueva métrica, se obtuvieron buenos patrones de convergencia, ver figura 41, donde se presenta una disminución de un 80% en el valor promedio de las evaluaciones al llegar a la generación N° 35. En cuanto al resultado final de la búsqueda se obtuvo una precisión promedio de los parámetros de búsqueda en las 3 series estudiadas de 7.6%.

Además se implementó con éxito la nueva representación basada en splines de interpolación entre 6 puntos de control, asimismo pudo incorporarse exitosamente esta nueva representación a un robot en el entorno de simulación.

Finalmente realizo con éxito una búsqueda con algoritmos genéticos utilizando como espacio de búsqueda coordenadas de algunos de los 6 puntos de control del solido amorfo. Se realizo una búsqueda con un parámetro y posteriormente con 4 parámetros.

En el caso de 1 parámetro de búsqueda se obtuvo un resultado final con un 0.7% de error con respecto al robot real. Para el caso de 4 parámetros de búsqueda el error promedio obtenido fue de 1%.

En ambos casos se observó buena convergencia de la evaluación promedio, ver figuras 50 y 54 y buena monotonía en los patrones de evaluación figuras 49 y 53.

## 6 Bibliografía

- [1] J.C. Zagal, J. Ruiz-del-Solar, A.G. Palacios.(2008) Fitness based identification of a robot structure, in: Artificial Life XI: Proceedings of the Eleventh International Conference on the Simulation and Synthesis of Living Systems, MIT Press, Cambridge, pages 733\_741.
- [2] Bongard, Victor Zykov, Hod Lipson. (2006) Resilient Machines Through Continuous Self-Modeling, *science* 314 (5802):1118-1121.
- [3] Bongard, J. and Lipson, H. (2004). Once more unto the breach: Co-evolving a robot and its simulator. In *Proc. of the Ninth Int. Conf. on the Simulation and Synthesis of Living Systems (ALIFE9)*, pages 57.62.
- [4] P. Parada and J.C. Zagal. (2010). Robot Self Modeling using the Minimum Description Length Principle, Electrical Engineering Department, University of Chile, Chile.
- [5] Bongard, J. and Lipson, H. (2005). Nonlinear system identification using coevolution of models and tests, *IEEE Trans. Evol. Comput.*, vol. 9, pages 361–384.
- [6] J.C. Zagal, J. Delpiano, and J. Ruiz-del-Solar. (2009). Self-Modeling in Humanoid Soccer Robots. *Robotics and Autonomous Systems (Special Issue on Humanoid Soccer Robots)*, 57, pages 819-827.
- [7] Hiller, J. and H. Lipson (2009). Multi Material Topological Optimization of Structures and Mechanism. Proceedings of the 11th Annual conference on Genetic and evolutionary computation, pages 1521-1528.
- [8] J.C. Zagal, J. Ruiz-del-Solar. (2004). Learning to kick the ball using Back-to-Reality, RoboCup 2004: Robot Soccer World Cup VII, Lecture Notes in Computer Science, 3276, S, pages 335\_346.
- [9] Zagal, J.C., Ruiz-del-Solar, J., Vallejos.(2004). Back to Reality: Crossing the Reality Gap in Evolutionary Robotics. Proceedings of the IAV 2004, 5th IFAC Symposium on Intelligent Autonomous Vehicles.
- [10] J.C. Bongard and H. Lipson. (2004). Automated damage diagnosis and recovery for remote robotics. *IEEE Intl Conf. on Robotics and Automation (ICRA)*, pages 3545-3550
- [11] J. C. Bongard and H. Lipson. (2004). Automated robot function recovery after unanticipated failure or environmental change using a minimum of hardware trials, *Proc. NASA/DoD Conf. Evolvable Hardware*, Seattle, WA, 2004, pp. 169–176.
- [12] N. A. Mirza, C. L. Nehaniv, K. Dautenhahn, and R. (2007). Grounded sensormotor interaction stories in an information theoretic metric space for robot ontogeny, *J. Adapt. Behav.*, vol. 15, pages 167–187.
- [13] Hearn, D., Baker, M.P. (2004). Computer Graphics with OpenGL. 3th Edition. Prentice Hall.
- [15] V.B. Anand. (1993). Computer Graphics and Geometric Modeling for Engineers, Wiley, New York.
- [16] Goldberg, D. E. (1989). Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley Publishing Co., Inc., RedwoodCity, CA.
- [17] Jay Hilyard and Stephen Teilhet. (2006). C# Cookbook. O'Reilly.