Bloque VI

EJEMPLO DE APLICACIÓN

Índice

- Problema a resolver.
- Propuesta de uso de aprendizaje.
- Algoritmo desarrollado.
- Selección del clasificador.
- Resultados experimentales.
  - Entorno de simulación.
  - Montaje experimental.

PROBLEMA A RESOLVER

Problema a resolver (I)

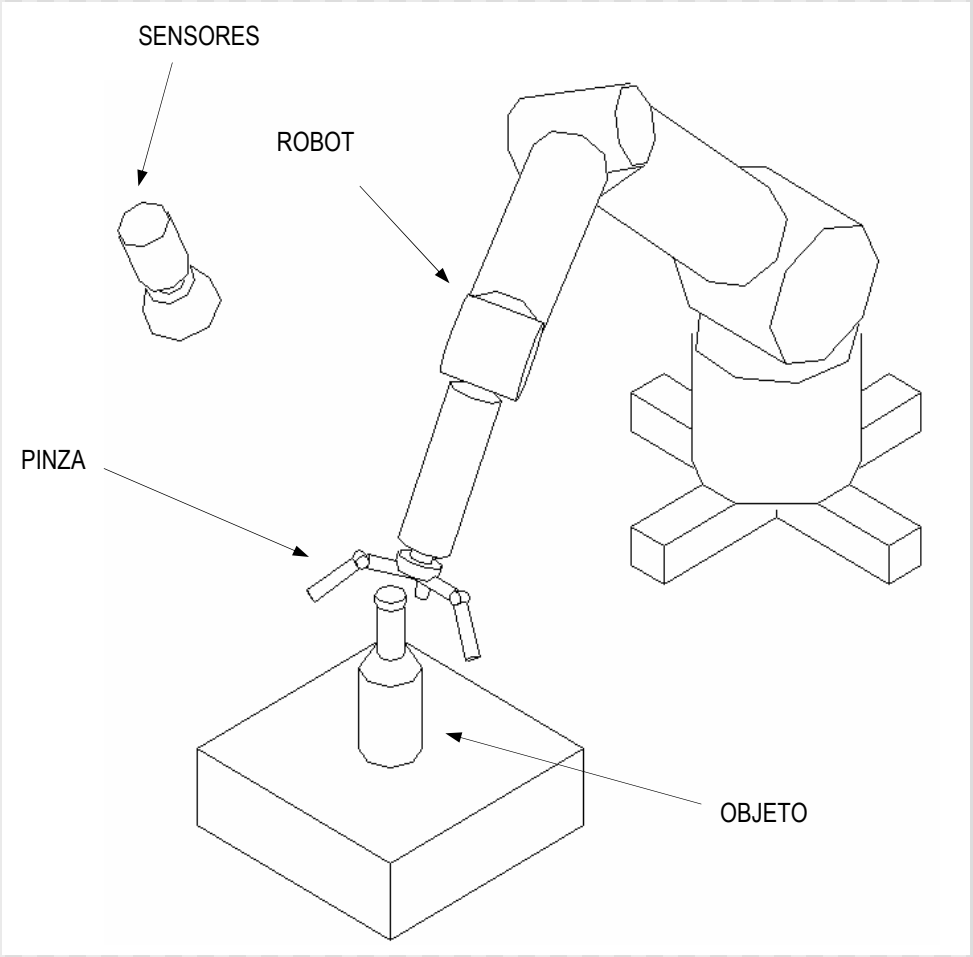
- Agarre de piezas mediante robots.
  - El objetivo es que un robot sea capaz de determinar el mejor modo de agarrar un objeto cualquiera.
  - Este problema podría ser resuelto sin utilizar aprendizaje automático.
  - No obstante, utilizar aprendizaje puede ofrecer ventajas.
  - Se intentará que el robot extraiga reglas de agarre a partir de ejemplos dados por el usuario.

Problema a resolver (II)

Problema a resolver (III)

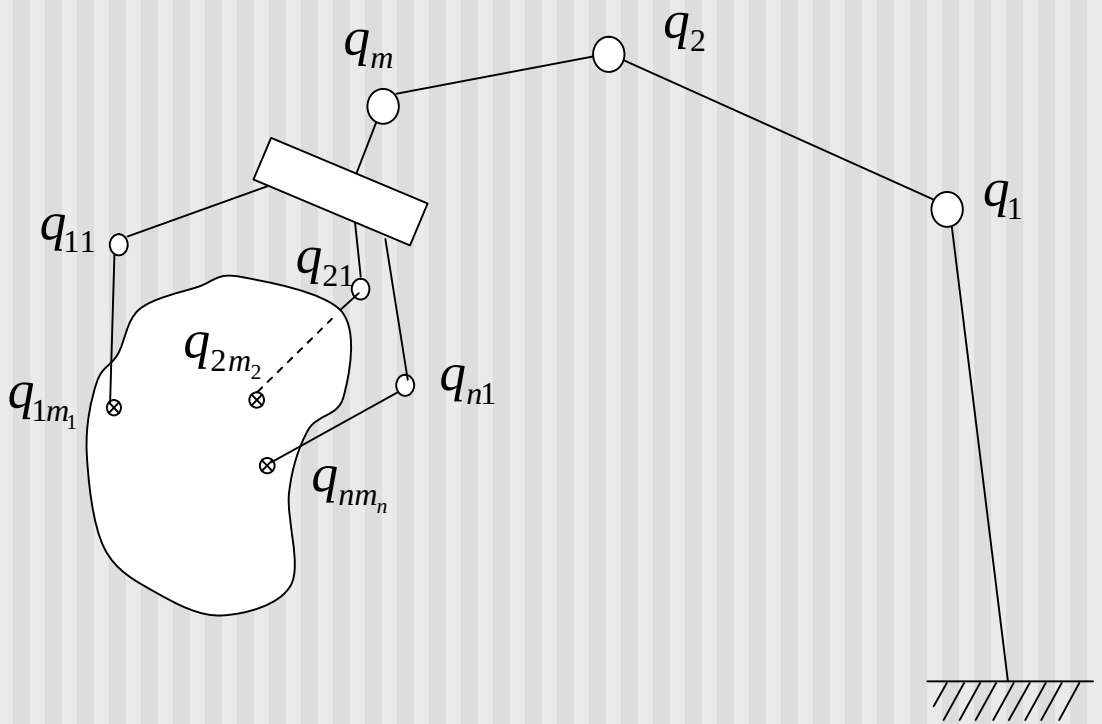
- Objetivo:
  - Cálculo del agarre óptimo para un objeto, dados un brazo robot y una pinza.

- Condicionantes:
  - Sólo se conoce la información proporcionada por los sensores (no un modelo 3D del objeto).
  - El agarre se debe definir completamente:
    - Puntos de contacto.
    - Configuración brazo y pinza (normalmente existe redundancia).



Problema a resolver (IV)

- Robot y pinza genéricos:
  - Brazo robot de *m* grados de libertad.
  - Pinza de *n* dedos cada uno de ellos con m_i grados de libertad.



PROPUESTA DE USO DE APRENDIZAJE

Propuesta aprendizaje (I)

- Se dispone de ejemplos de agarre realizados por el usuario en teleoperación.
- La calidad de un nuevo agarre equivale a su similitud a los ejemplos.
- La similitud se calcula a partir de datos locales y globales.
  - Datos locales: afectan a cada punto de contacto por separado (ej. curvatura).
  - Datos globales: indican la situación relativa de los puntos de contacto (ej. cierre de fuerza o force closure).

Propuesta aprendizaje (II)

- Datos locales (referidos a cada punto de contacto):
  - Distancia al centro de gravedad.
  - Medida multirresolución de la curvatura 2D o 3D.
- Datos globales (referidos al conjunto de todos los puntos de contacto):
  - Se establece un punto de referencia: el centro del espacio convexo definido por el conjunto de los puntos de contacto. A partir de él se obtienen 2 medidas:
  - Distancia desde el punto de referencia al centro de gravedad de la pieza (permite distinguir los agarres centrados de los que no lo son).
  - Medida multirresolución del ángulo formado por la normal a la superficie en el punto de contacto y la línea dirigida hacia el punto de referencia (guarda relación con el cumplimiento de la condición de cierre de fuerza o force closure).

Propuesta aprendizaje (III)

¿Por qué se separan las medidas locales y globales?

- Permite reducir el tiempo de cómputo:
  - Primera fase: se estudian los puntos de la superficie utilizando atributos locales.
  - Segunda fase: se estudian sólo las posibles combinaciones de puntos válidos utilizando atributos globales.
  - Resultado: reducción de tiempo de cómputo muy elevada, particularmente en el caso de pinzas de múltiples dedos.

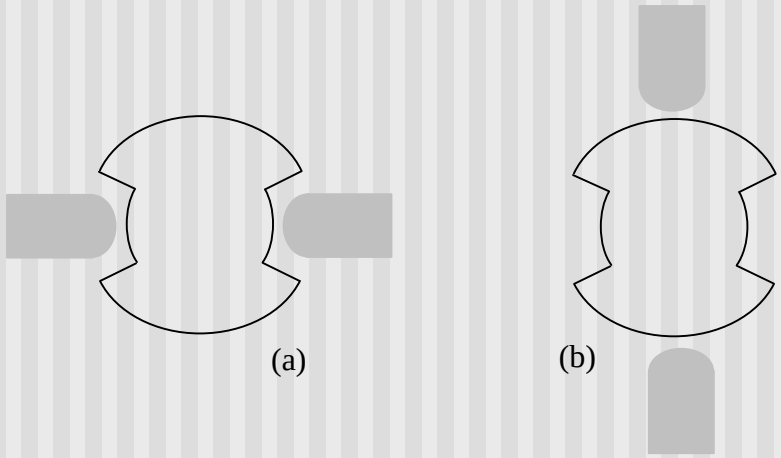
$$r_{reducc} \approx \left(\frac{z}{z'} \right)^n$$

- Permite que el sistema de síntesis de agarres propuesto sea más genérico:
  - Válido para diferentes pinzas.
  - Válido para diferentes tipos de información sensorial.

Propuesta aprendizaje (IV)

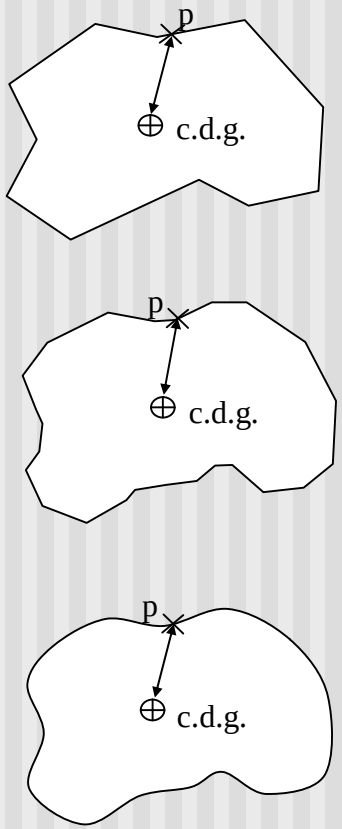
¿Por qué se utilizan medidas multirresolución?

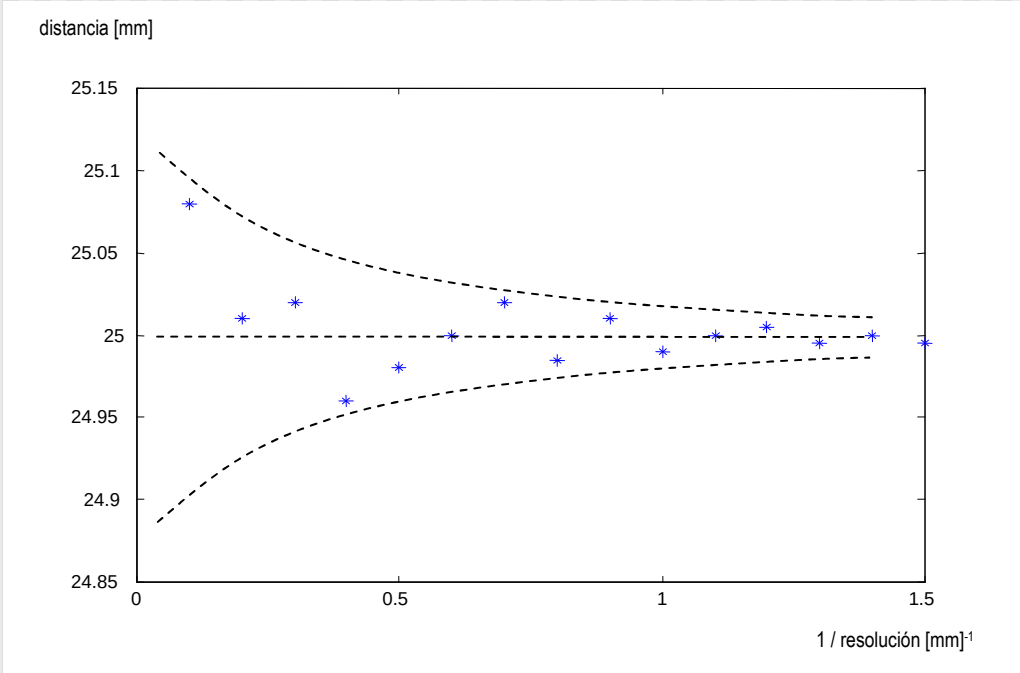
- Ciertos atributos ofrecen información dependiente de la resolución:
  - Atributos de distancia: convergen hacia un valor al aumentar la resolución: basta con elegir la máxima resolución disponible.
  - Atributos angulares (concavidades, direcciones de la normal a la superficie en el punto de contacto, etc.): no convergen al aumentar la resolución.
- Utilizar medidas multirresolución permite diferenciar agarres que de otro modo serían indistinguibles.



Propuesta aprendizaje (V)

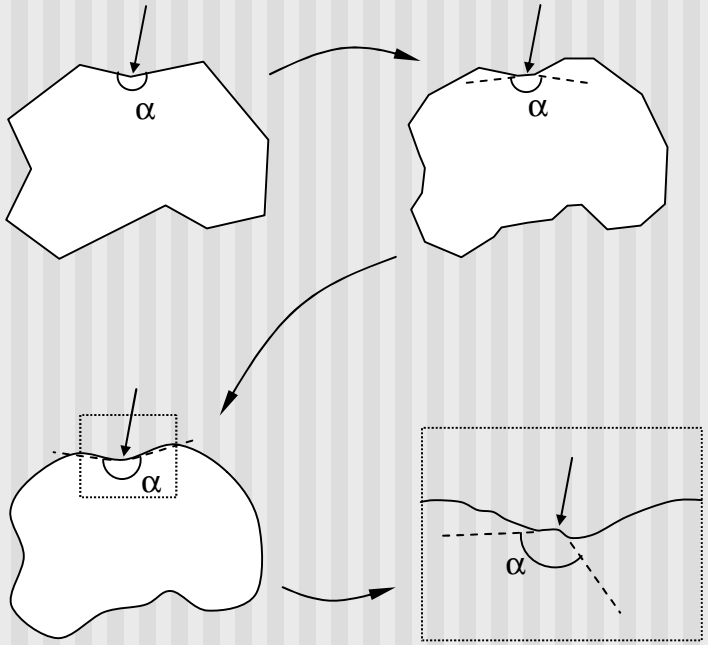
- Convergencia de una medida de distancia:

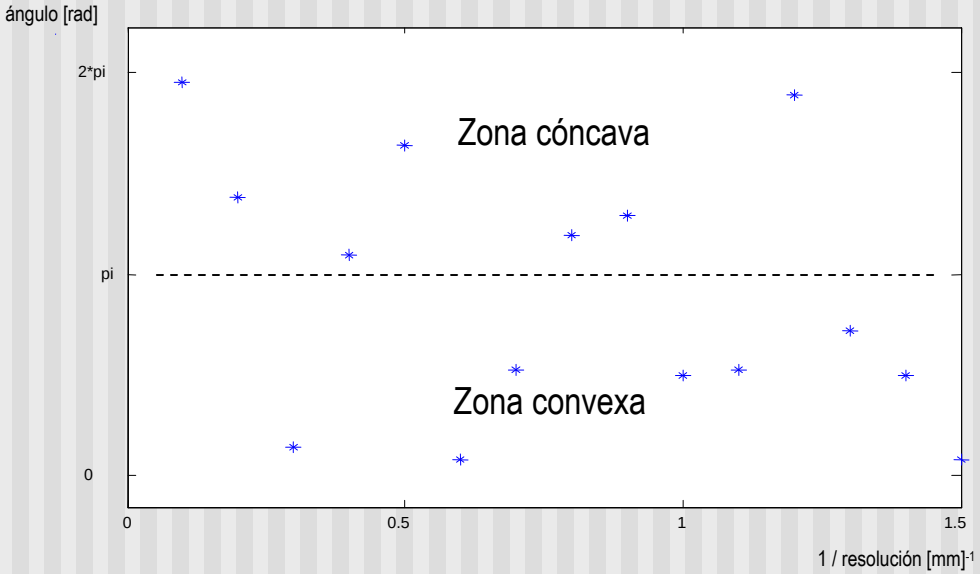




Propuesta aprendizaje (VI)

- No convergencia de una medida angular:





ALGORITMO

Algoritmo (I)

- La síntesis de un agarre se realiza en cuatro etapas:

1. Filtrado de los puntos de la superficie del objeto.

$$F = \{p_1, p_2, \ldots, p_z\} \in \Re^k \quad F' = \{p_1, p_2, \ldots, p_{z'}\} \in \Re^k : p_i \text{ válido } \forall i \quad F' \subset F \quad z' \leq z$$

2. Cálculo de conjuntos de *n* puntos de contacto (*n* dedos) alcanzables por la pinza.

$$G = \{g_1, g_2, \ldots, g_s\} \quad g_i = \{p_1, p_2, \ldots p_n\} : p_i \text{ válido } \forall i \, , p_i \neq p_j \; \forall i \neq j$$

$$G' = \{g_1, g_2, \ldots, g_{s'}\} : g_i \text{ alcanzable } \forall i \quad G' \subset G \quad s' \leq s$$

3. Selección del conjunto óptimo de puntos.

$$G'' = (\{g_1, g_2, \ldots, g_{s''}\} : g_i \text{ alcanzable } \forall i \, , g_i \text{ válido } \forall i) \quad G'' \subset G' \quad s'' \leq s'$$

$$g_{selecc} = g_i \in G'' : \min_{\forall k}(dist(g_i, ej_k)) \leq \min_{\forall k}(dist(g_j, ej_k)) \forall j \neq i$$

4. Selección de la configuración óptima para la pinza (si existe redundancia).

$$Q_i = \{q_1, \ldots, q_m, q_{11}, \ldots, q_{1m_1}, \ldots, q_{n1}, \ldots, q_{nm_n}\} : f(Q_i) = g_{selecc} \; \forall i$$

$$Q_{selecc} = Q_i : \min_{\forall k}(dist(q_i, q_k^{ej})) \leq \min_{\forall k}(dist(q_j, q_k^{ej})) \forall j \neq i$$

Algoritmo (II)

1 $F = \{p_1, p_2, \ldots, p_z\} \in \Re^k$ $F' = \{p_1, p_2, \ldots, p_{z'}\} \in \Re^k : p_i \text{ válido } \forall i$ $F' \subset F$ $z' \leq z$

- Objetivo: decidir cuáles de los puntos de la superficie de la pieza son válidos para situar sobre ellos un dedo de la pinza.
- Se utiliza aprendizaje automático: la validez se evalúa como similitud a los ejemplos de entrenamiento.
- Atributos considerados: MEDIDAS LOCALES.
- Se utiliza aprendizaje inductivo.
  - Estudio de la aplicabilidad de diferentes métodos:

<table>
<tr><td>Lazy</td><td colspan="2">Vecino más cercano (k-NN)</td></tr>
<tr><td rowspan="3">Eager</td><td>Perceptrón multicapa</td><td>Funciones de base radial</td></tr>
<tr><td colspan="2">Naive Bayes</td></tr>
<tr><td>Árboles de decisión (C4.5)</td><td>Listas de reglas (PART)</td></tr>
</table>

  - Conclusión: se utilizan árboles de decisión.

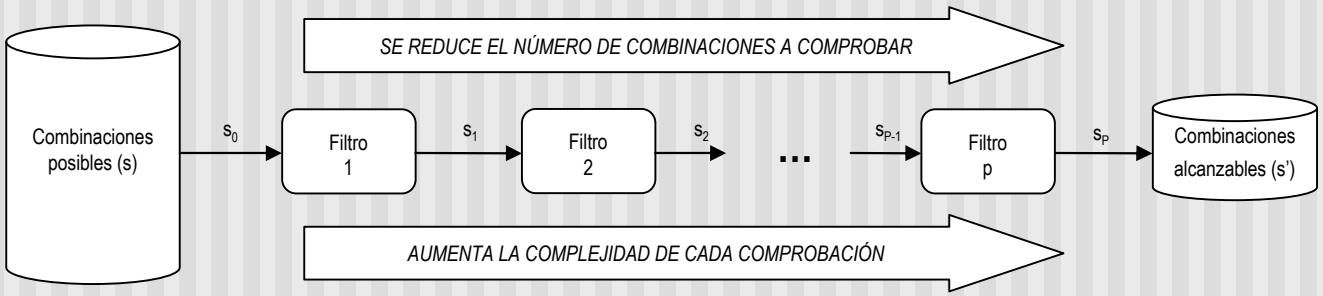
Algoritmo (III)

2

$G = \{g_1, g_2, \ldots, g_s\}$ $g_i = \{p_1, p_2, \ldots p_n\}$: p_i *válido* $\forall i$, $p_i \neq p_j$ $\forall i \neq j$

$G' = \{g_1, g_2, \ldots, g_{s'}\}$: g_i *alcanzable* $\forall i$ $G' \subset G$ $s' \leq s$

- Objetivo: calcular todas las posibles combinaciones de n puntos de contacto válidos alcanzables por la pinza.
- Número de agarres a estudiar: $\binom{z'}{n}$.
- Posibles causas de no alcanzabilidad de un conjunto de puntos:
  - Limitaciones cinemáticas de brazo robot y pinza.
  - Existencia de colisiones.
- Estudio completo inviable computacionalmente.
- Propuesta: filtros sucesivos de complejidad creciente.



Algoritmo (IV)

3

$$G'' = \left(\{g_1, g_2, \ldots, g_{s''}\} : g_i \text{ alcanzable } \forall i \, , g_i \text{ válido } \forall i\right) \quad G'' \subset G' \quad s'' \leq s'$$

$$g_{selecc} = g_i \in G'' : \min_{\forall k}\left(dist\left(g_i, ej_k\right)\right) \leq \min_{\forall k}\left(dist\left(g_j, ej_k\right)\right) \forall j \neq i$$

- Objetivo: elegir la combinación de *n* puntos de contacto más adecuada.
- De nuevo se utiliza aprendizaje automático: la validez se evalúa como *similitud a los ejemplos de entrenamiento*.
- Atributos considerados: MEDIDAS GLOBALES
- Se procede en dos pasos:
  - Filtrado de las combinaciones inválidas.
  - Selección de la combinación más apropiada de entre las válidas.
- Se compara el funcionamiento de diversos métodos de aprendizaje, seleccionando:
  - Primer paso: árbol de decisión.
  - Segundo paso: variación del método del vecino más cercano.

Algoritmo (V)

4

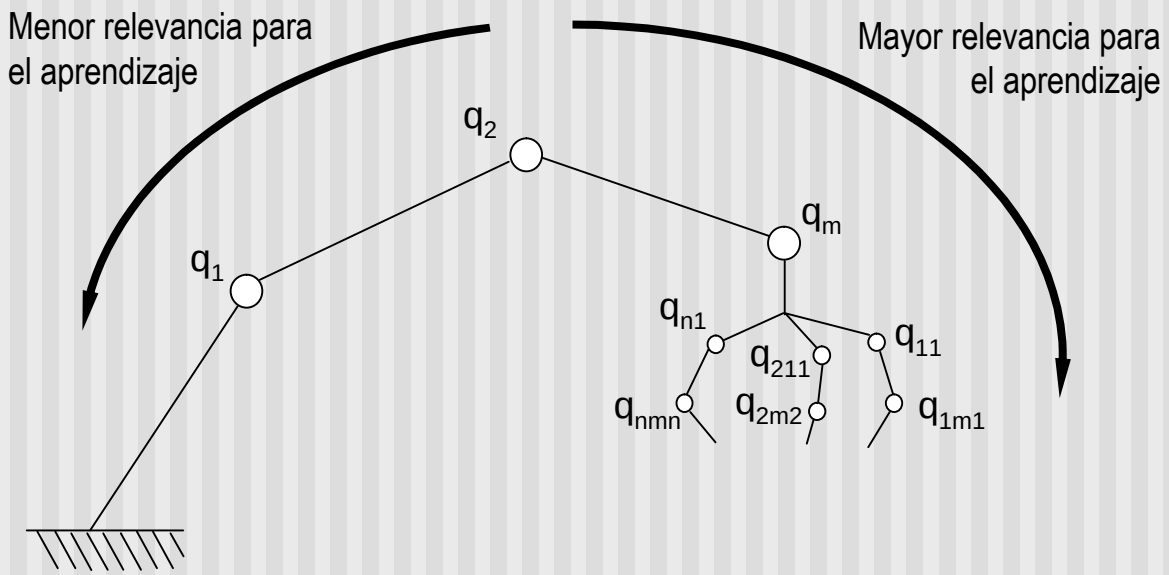
$$Q_i = \{q_1, ..., q_m, q_{11}, ..., q_{1m_1}, ..., q_{n1}, ..., q_{nm_n}\} : f(Q_i) = g_{selecc} \ \forall i$$

$$Q_{selecc} = Q_i : \min_{\forall k}\left(dist\left(q_i, q_k^{ej}\right)\right) \leq \min_{\forall k}\left(dist\left(q_j, q_k^{ej}\right)\right) \forall j \neq i$$

- Objetivo: seleccionar una configuración de entre todas las que alcanzan los puntos de contacto.
- Se trata de un problema de resolución de redundancia cinemática.
- Propuesta basada en tres conceptos fundamentales:
  - Resolución de la redundancia en función de la similitud con los ejemplos de entrenamiento.
  - Ponderación de las distintas articulaciones de brazo y pinza robot en función de su influencia en la calidad del agarre.
  - Representación de la cinemática inversa en forma de árbol.
- Resultado: procedimiento computacionalmente eficiente.

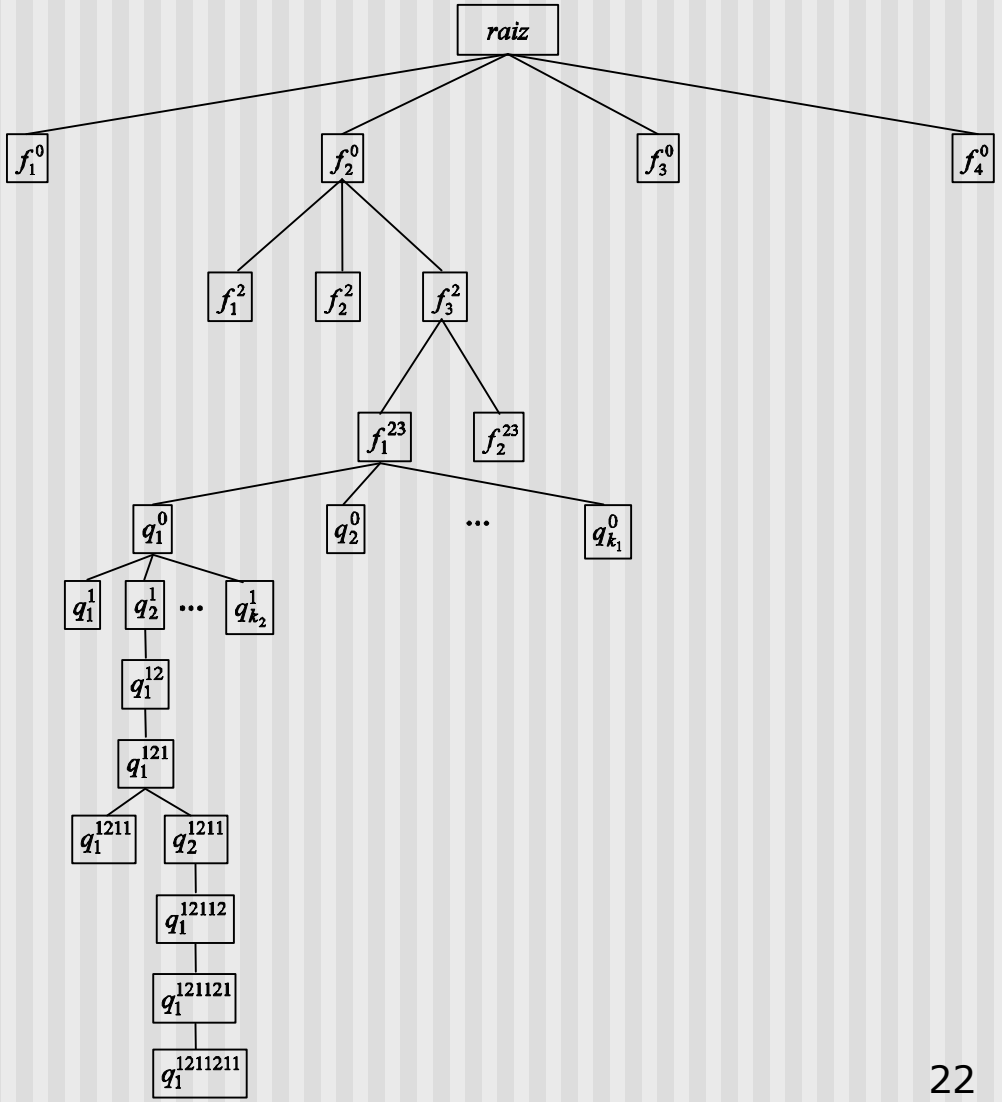
Algoritmo (VI)

- Resolución redundancia: similitud con las configuraciones de los ejemplos.
- Diferencia entre dos configuraciones: ponderada en función de las articulaciones.
- Criterio genérico de ponderación: cercanía a los puntos de contacto con el objeto.
  - Articulaciones próximas: indican el tipo de agarre realizado.
  - Articulaciones alejadas: dependen fundamentalmente de la posición del objeto respecto del robot.
- Criterio modificable según necesidades.



Algoritmo (VII)

- Árbol cinemático:
- Refleja todas las posibles soluciones de la cinemática inversa a un cierto nivel de resolución (discretización).
- Priorización:
  - n-1 primeros niveles: asignaciones de los n dedos.
  - restantes niveles: articulaciones de brazo y pinza.
- Computacionalmente eficiente:
  - Búsqueda heurística sin vuelta atrás.
  - Construcción y búsqueda simultáneas, no es necesario construir el árbol completo.



Algoritmo (VIII)

- Se busca la configuración parecida a alguna de las utilizadas en los ejemplos para agarrar un objeto con atributos parecidos (dos medidas de distancia).
- Distancia en el espacio de características (similitud objetos):

$$d_F\left(S_p, S_q\right) = \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{f_L}\left(\hat{l}_j^{p_i} - \hat{l}_j^{q_i}\right)^2 + \sum_{j=1}^{f_G}\left(\hat{g}_j^{S_p} - \hat{g}_j^{S_q}\right)^2}$$

- Distancia en el espacio de configuración (similitud asignaciones / articulaciones):

$$d_C\left(p_a, p_b\right) = \sqrt{\sum_{i=1}^{f_L}\left(\hat{l}_i^{p_a} - \hat{l}_i^{p_b}\right)^2} \qquad d_J\left(q_i^a, q_i^b\right) = \left|q_i^a - q_i^b\right|$$

- Combinación ambos efectos: parámetro ajustable (sólo necesario en el primer nivel):

$$d_T\left(S_p, S_q, p_a, p_b\right) = a \cdot \frac{d_F\left(S_p, S_q\right)}{\sqrt{n \cdot f_L + f_G}} + \left(1 - a\right) \cdot \frac{d_C\left(p_a, p_b\right)}{\sqrt{f_L}}$$

- Siempre medidas normalizadas.

SELECCIÓN DEL CLASIFICADOR

Selección del clasificador (I)

- Métodos comparados:

Lazy	Vecino más cercano (k-NN)	
Eager	Perceptrón multicapa	Funciones de base radial
	Naive Bayes	
	Árboles de decisión (C4.5)	Listas de reglas (PART)

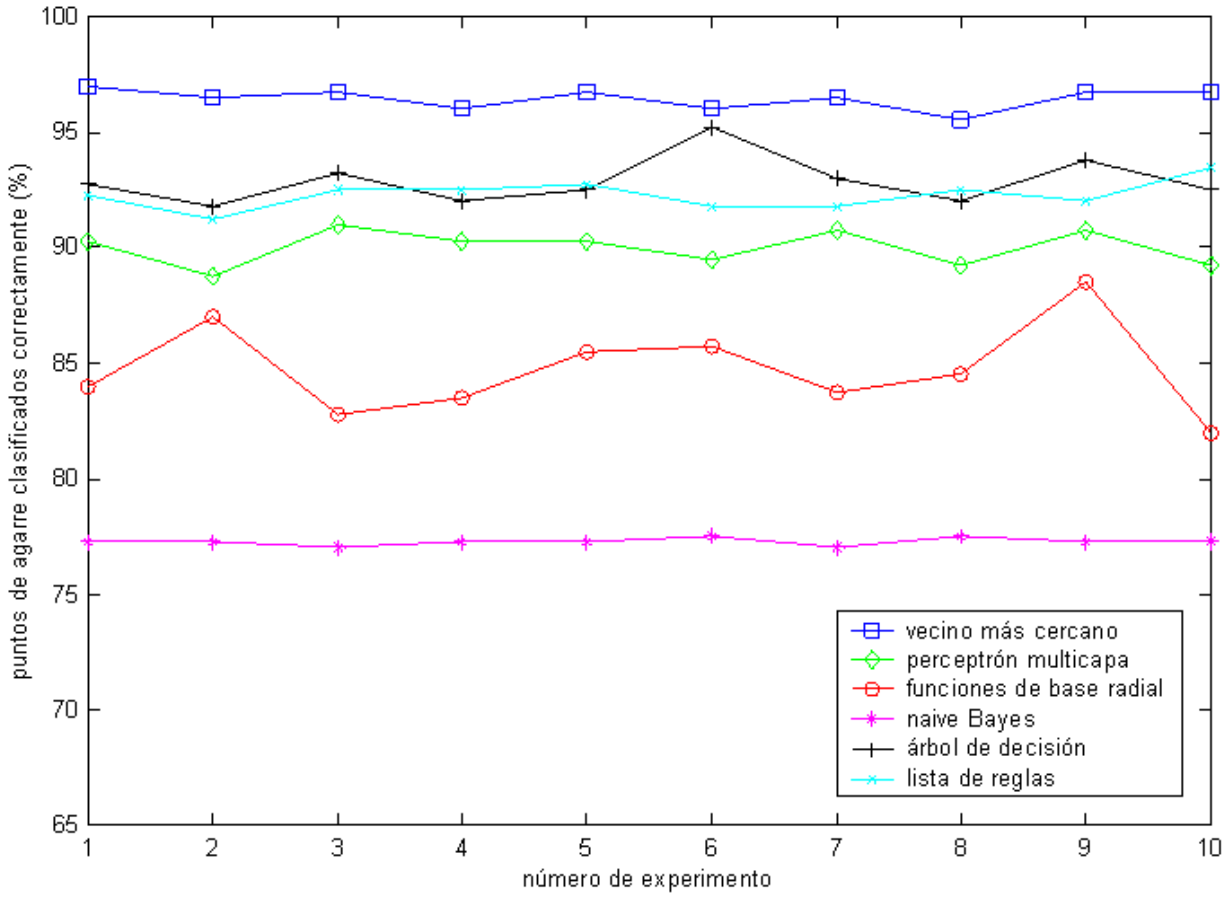
- Criterios de comparación empleados
  - Cuantitativos:
    - Validación cruzada:
      - 10 divisiones, selección estratificada.
      - Base de datos de 400 ejemplos (360 entrenamiento).
    - Porcentaje de clasificaciones correctas con número variable de ejemplos de entrenamiento:
      - Se detectan problemas de sobreajuste.
      - El número de ejemplos varía entre 20 y 380.
    - Tiempo de procesado, *off-line* y *on-line*.
  - Cualitativos:
    - Legibilidad de los resultados.
    - Robustez frente a ejemplos de entrenamiento ruidosos.
    - Sensibilidad a variaciones de los parámetros del algoritmo.

Selección del clasificador (II)

- Experimentos iniciales de ajuste de parámetros. Ajustes óptimos:
  - Vecino más cercano:
    - Sólo un vecino (1-NN).
  - Perceptrón multicapa:
    - Una capa oculta con 9 unidades (7 unidades para la etapa 3).
    - Tasa de aprendizaje: 0.3; momento 0.2; epochs: 500 (valores por defecto).
  - Funciones de base radial:
    - 40 funciones base (10 unidades para la etapa 3).
  - Naive Bayes:
    - Función de densidad de probabilidad estimada mediante suma de gaussianas.
  - Árboles de decisión (C4.5):
    - Nivel de confianza para la poda: C= 0.3. (0.25 para la etapa 3).
  - Listas de reglas (PART):
    - Nivel de confianza para la poda: C= 0.3. (0.15 para la etapa 3).

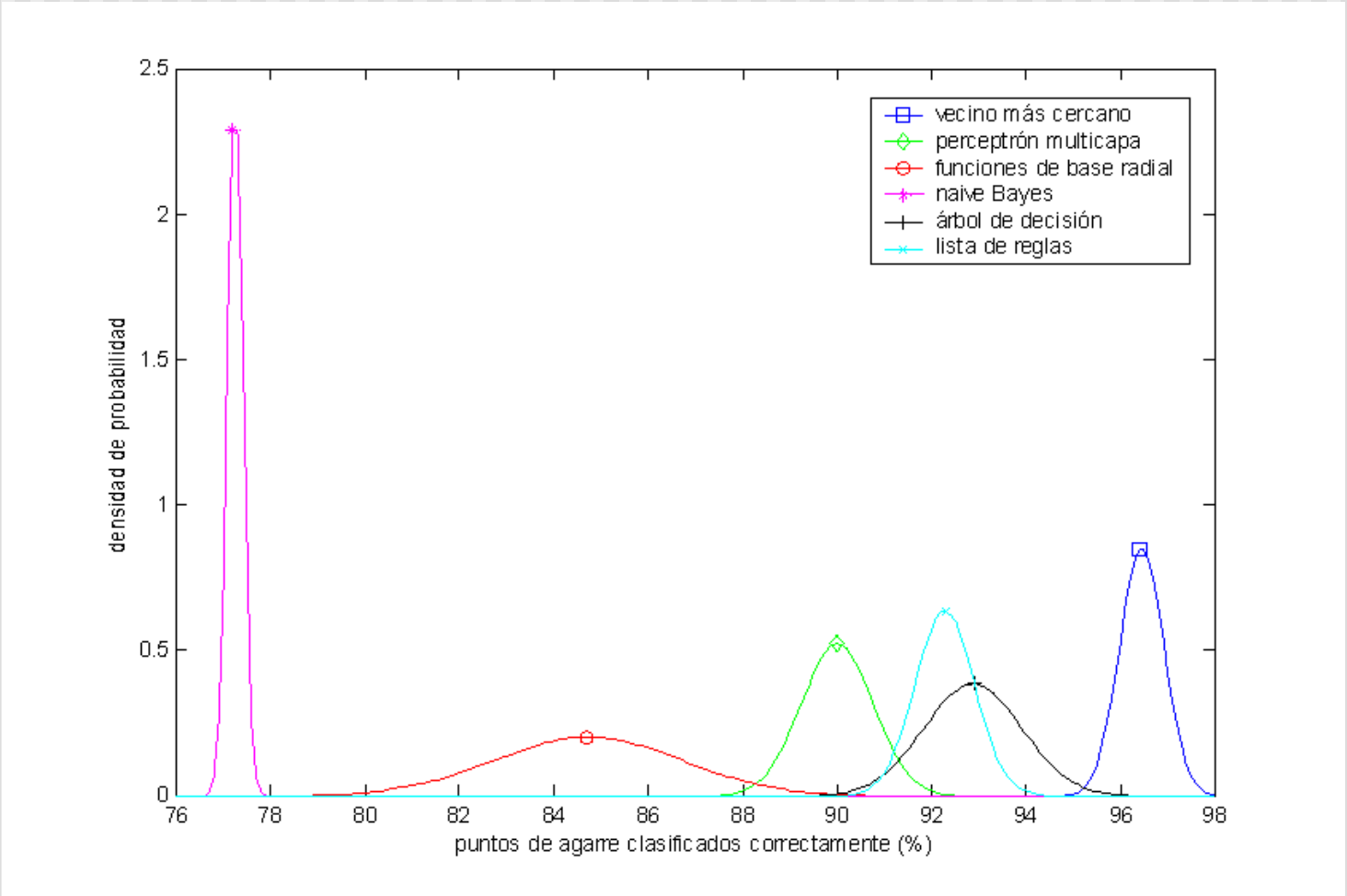
Selección del clasificador (III)

- Validación cruzada:



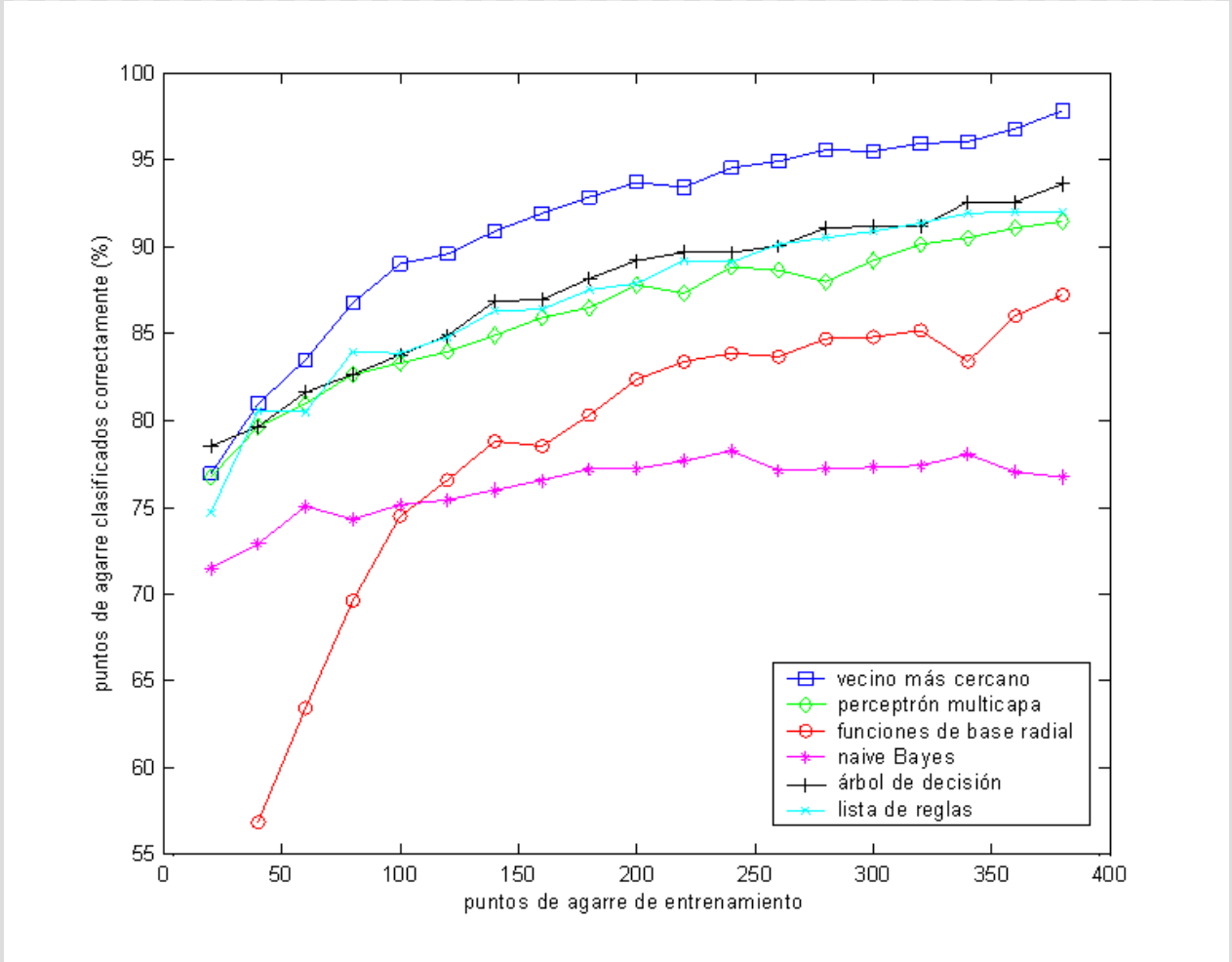
Selección del clasificador (IV)

- Validación cruzada (resultados ajustados a normal):



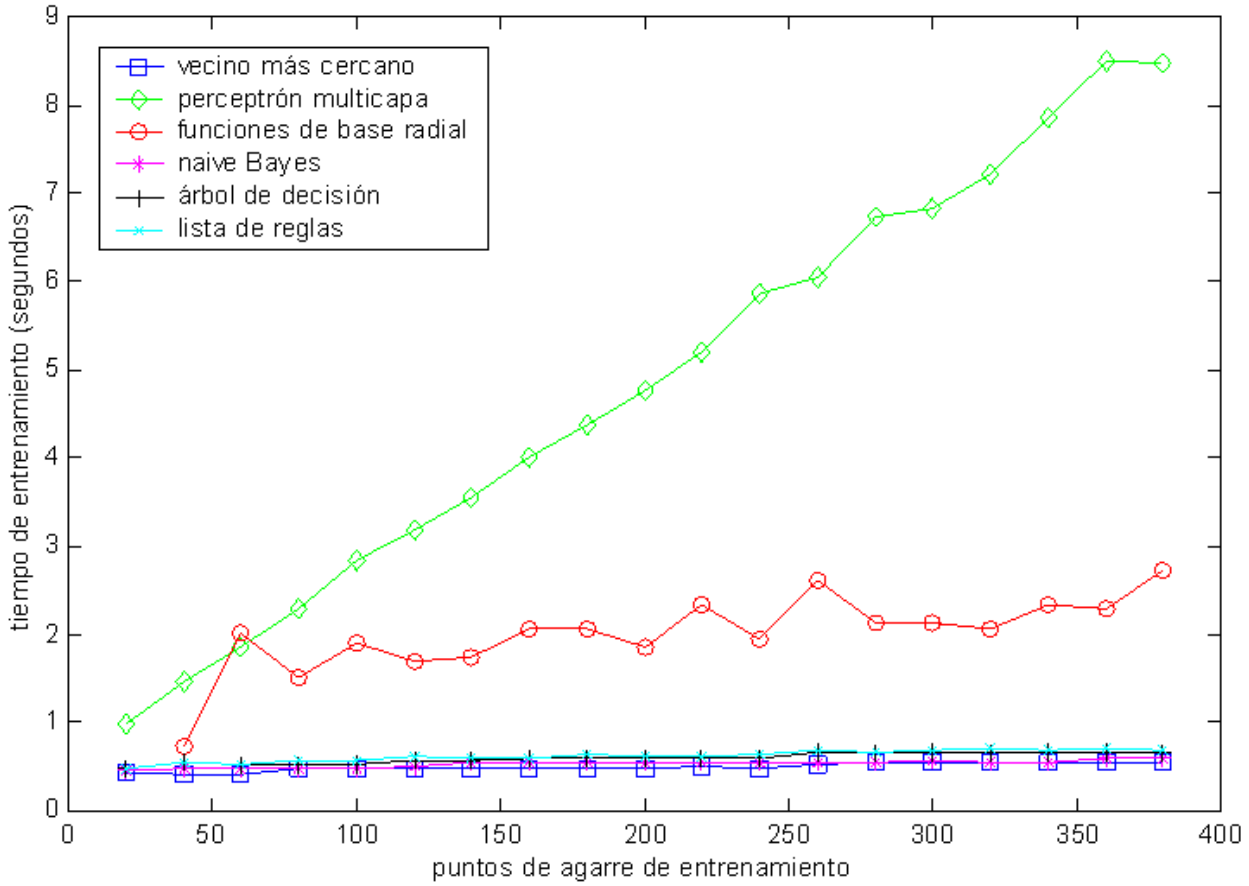
Selección del clasificador (V)

- Resultados con número variable de ejemplos:



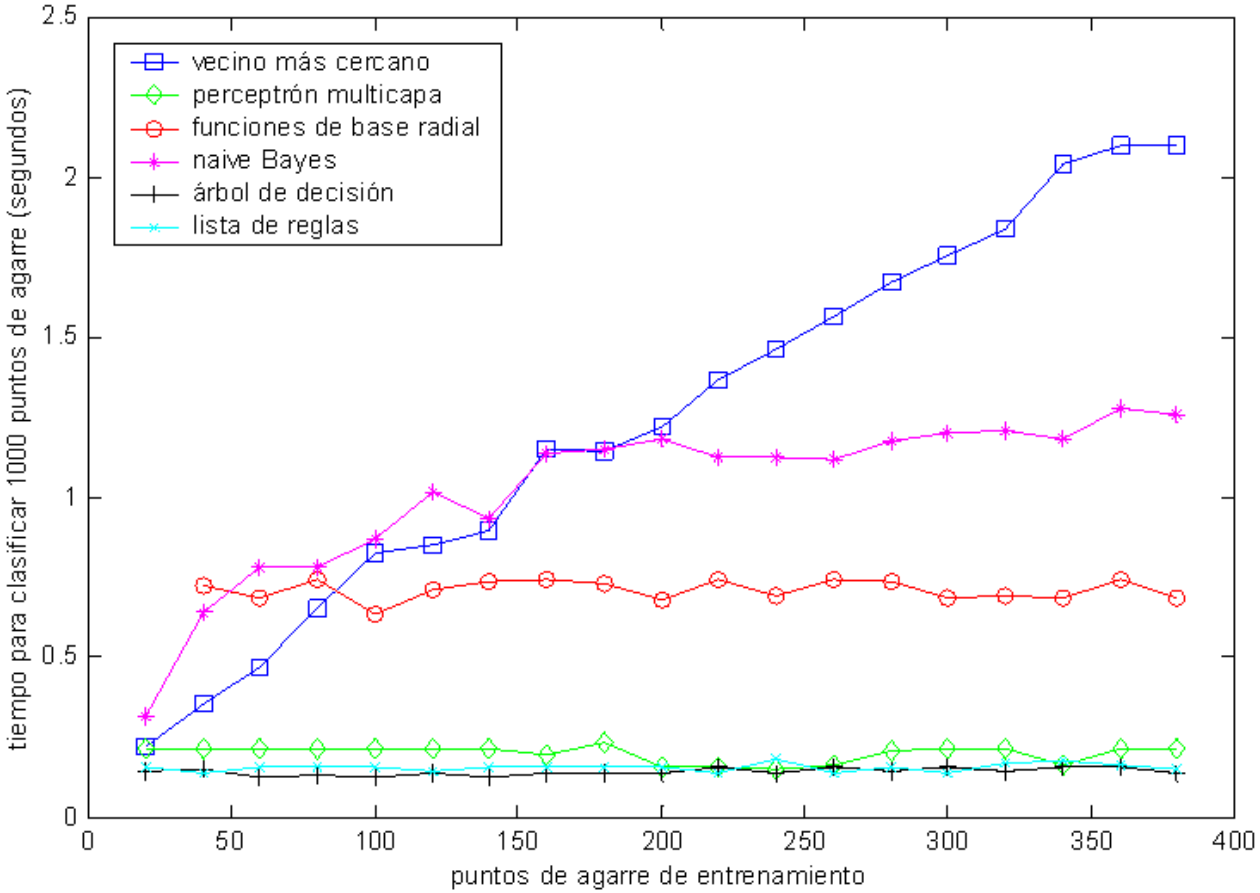
Selección del clasificador (VI)

- Tiempo de cómputo off-line:



Selección del clasificador (VII)

- Tiempo de cómputo on-line:



Selección del clasificador (VIII)

- Criterios cualitativos:

	Legibilidad del modelo	*Robustez ante ruido en ejemplos de entrenamiento*	*Robustez ante variaciones parámetros*
Árboles de decisión	X	X	X
Listas de reglas	X	X	X
Vecino más cercano			X
Perceptrón multicapa		X	
Funciones de base radial			X
Naive Bayes		X	X

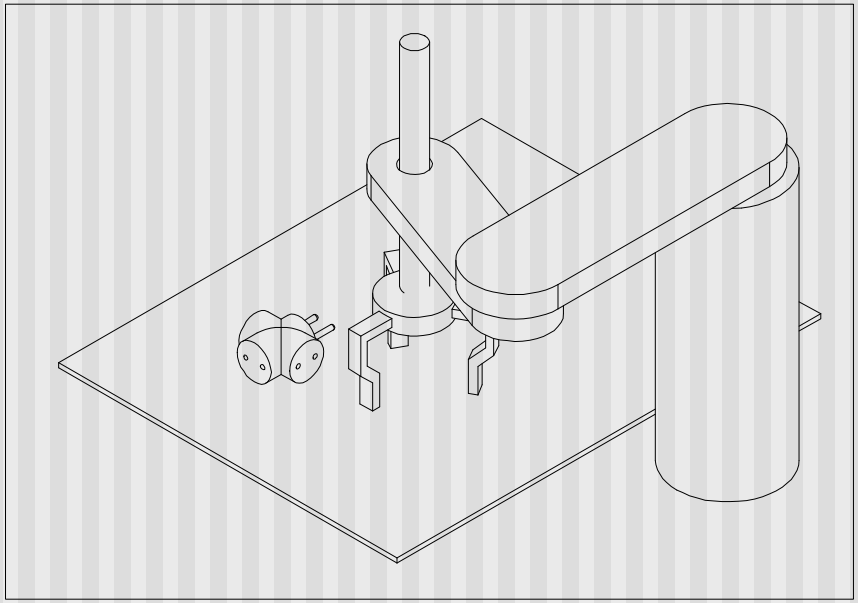
Selección del clasificador (IX)

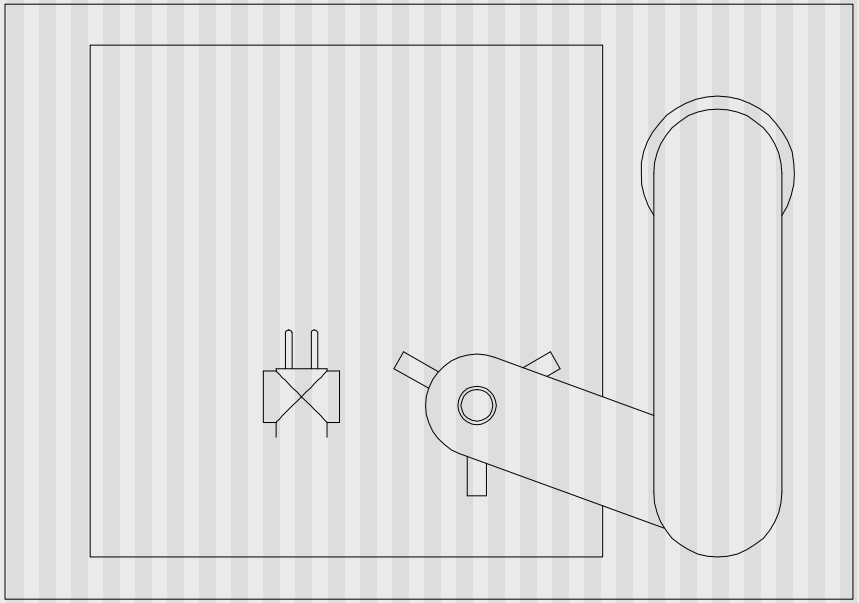
- Conclusiones:
- Los árboles de decisión y las listas de reglas son los mejores métodos de acuerdo con los criterios cualitativos y cuantitativos.
- Se eligen los árboles de decisión por su mayor legibilidad global (muestran atributos priorizados).
- Consideraciones:
  - El algoritmo de síntesis de agarres desarrollado permite la utilización de cualquier clasificador.
  - Los resultados mostrados corresponden a la primera fase del algoritmo de síntesis de agarre (clasificación de puntos en función de atributos locales). En la tercera etapa del algoritmo (clasificación de conjuntos en función de atributos globales) se obtienen resultados similares.

RESULTADOS EXPERIMENTALES

Resultados experimentales (I)

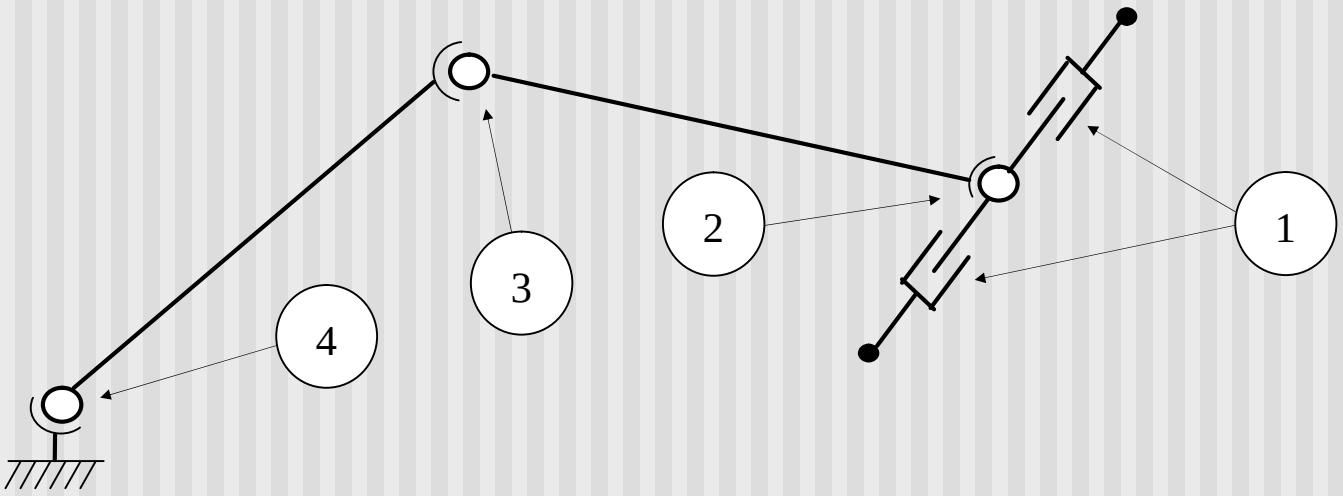
- Problema: agarre de una pieza sobre una superficie plana.
- Datos sensoriales: obtenidos por cámara cenital (contorno)
- Robot SCARA: planteamiento 2D desacoplando el eje z.





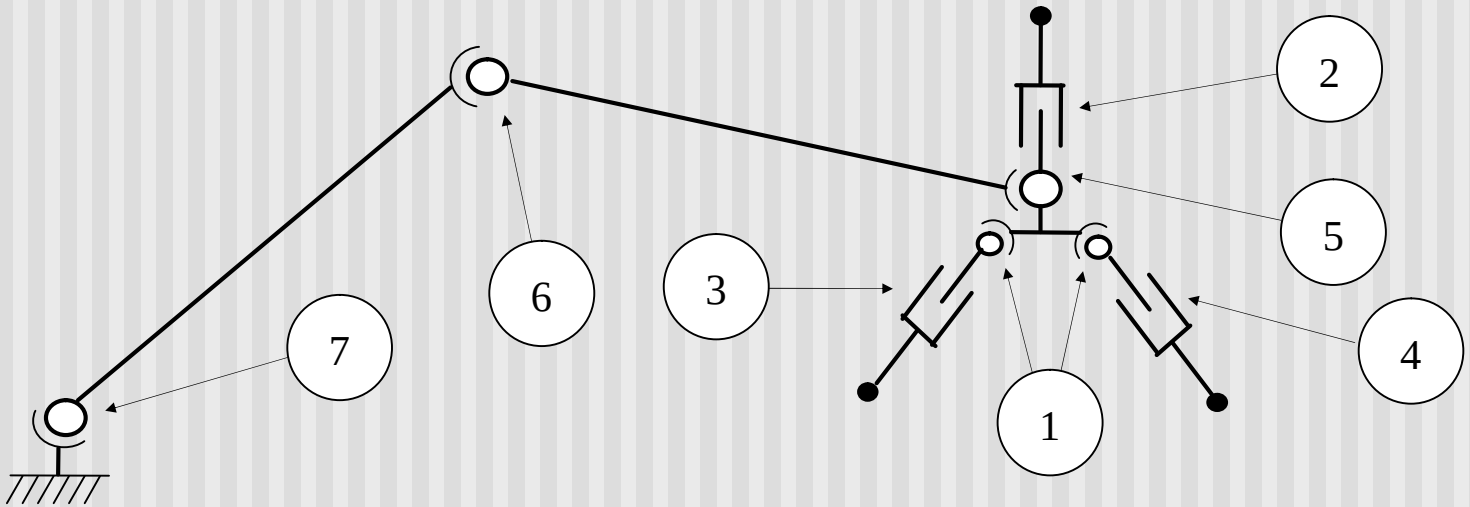
Resultados experimentales (II)

- Primera pinza: paralela de dos dedos.
- 2 puntos de contacto.
- Cierre simultáneo dedos.
- No existe redundancia cinemática (salvo codo arriba/abajo).
- 4 articulaciones priorizadas.



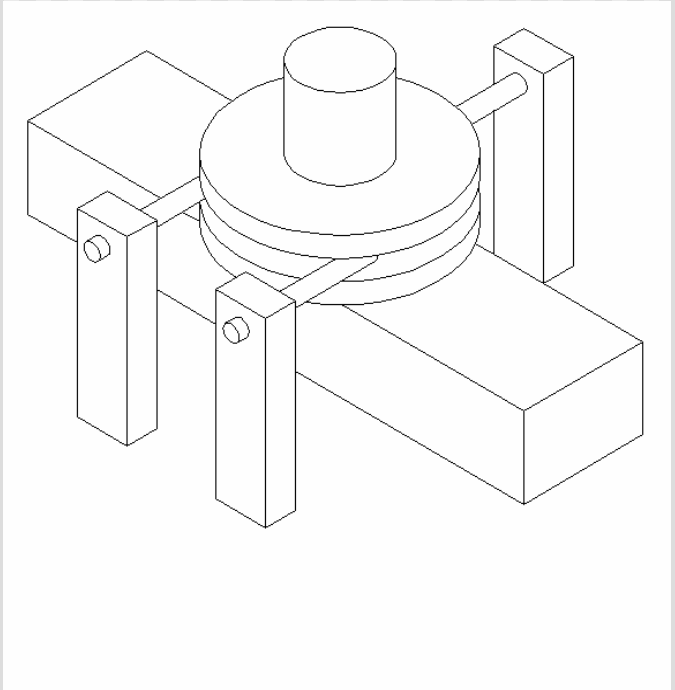
Resultados experimentales (III)

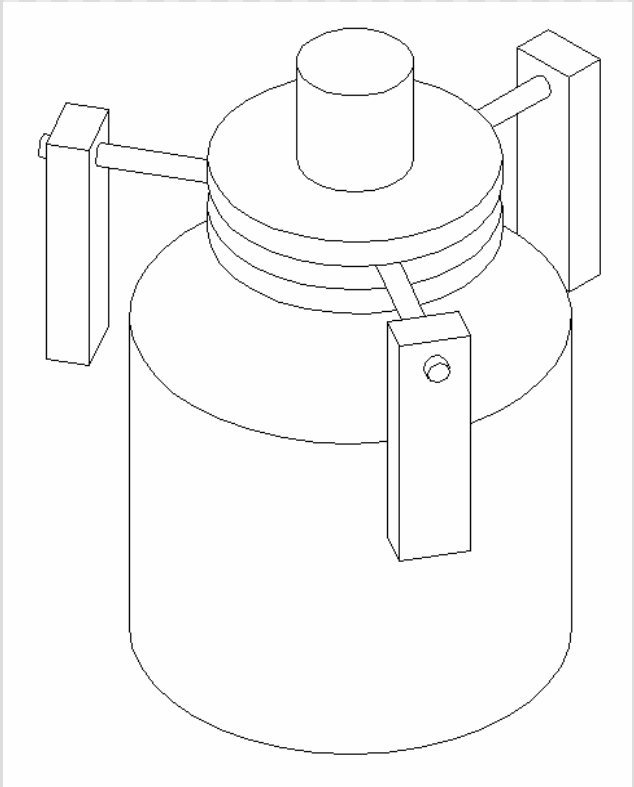
- Segunda pinza: articulada de tres dedos.
- 3 puntos de contacto
- Cierre independiente dedos, abducción simultánea.
- Existe una elevada redundancia.
- 7 articulaciones priorizadas.



Resultados experimentales (IV)

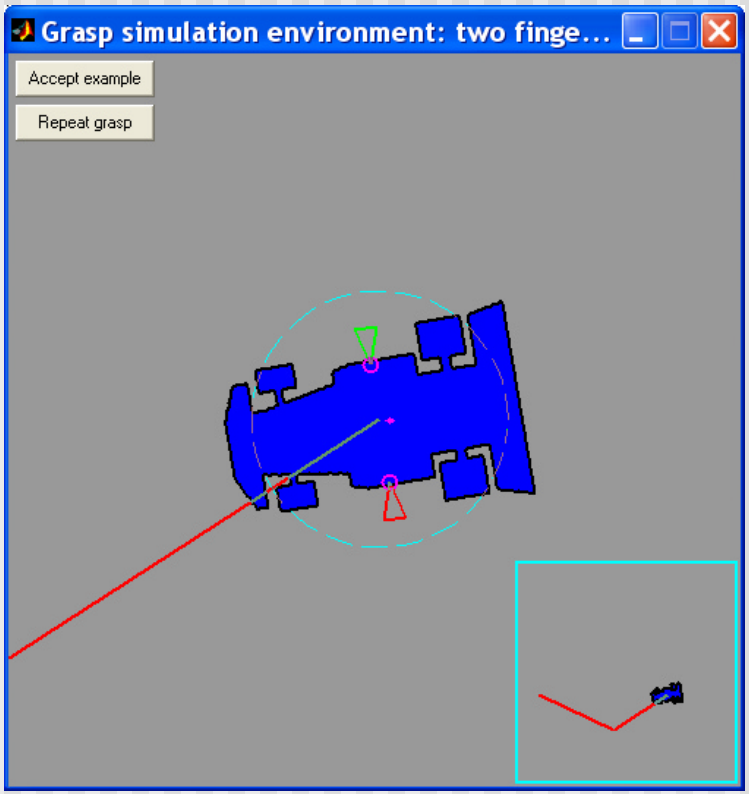
- Priorización articulaciones en segundo ejemplo:
  - Máxima prioridad para la abducción de dedos.
  - Permite distinguir tipos de agarres y asociarlos a objetos.

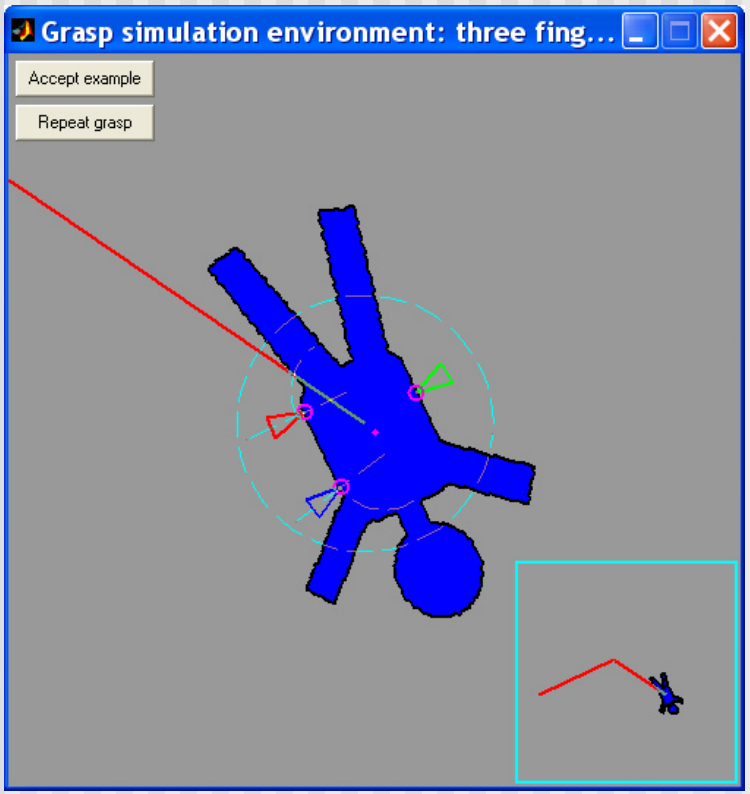




Resultados experimentales (V)

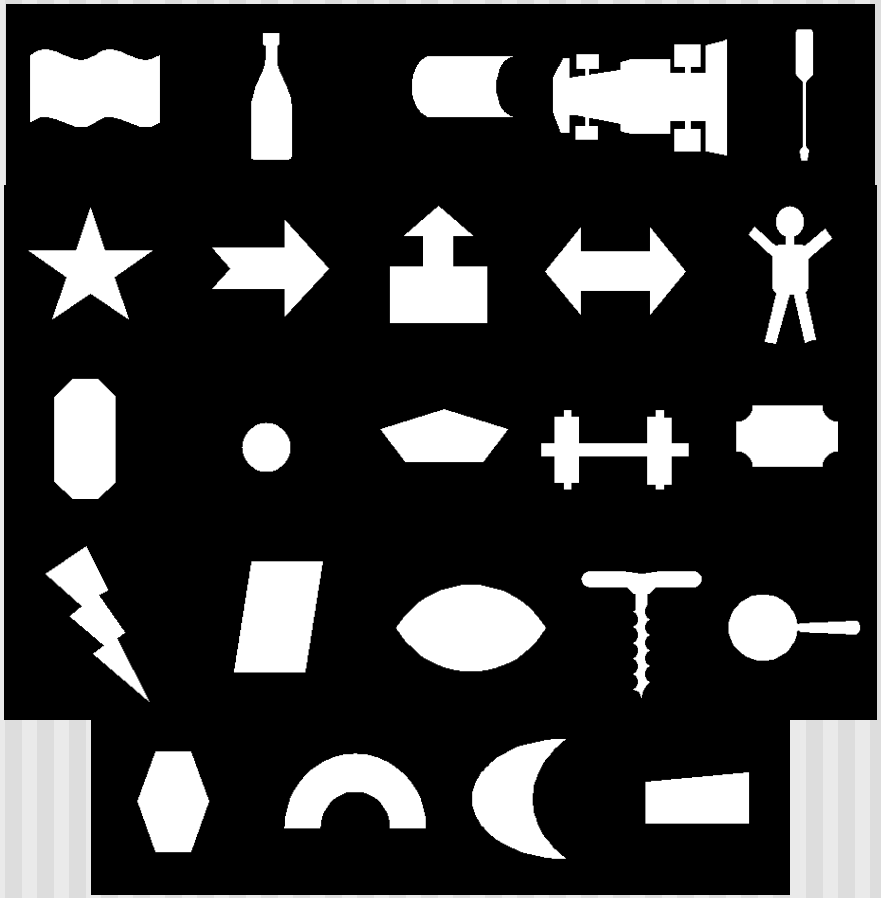
- Entorno de simulación: desarrollado en Matlab.
- Válido para entrenamiento y funcionamiento autónomo.
- Preparado para pinzas de dos y tres dedos.
- Permite comprobar el funcionamiento del algoritmo paso a paso.



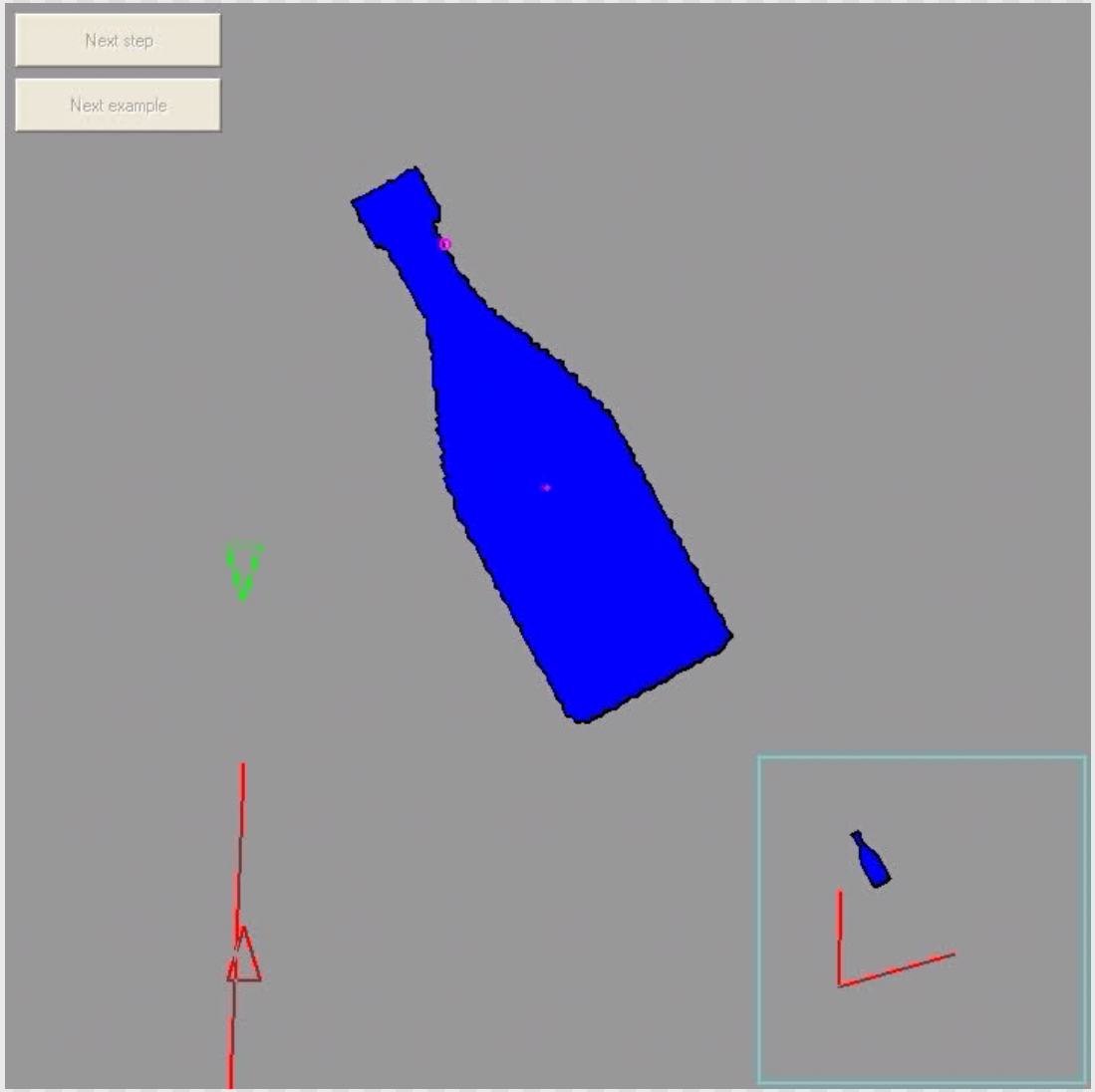


Resultados experimentales (VI)

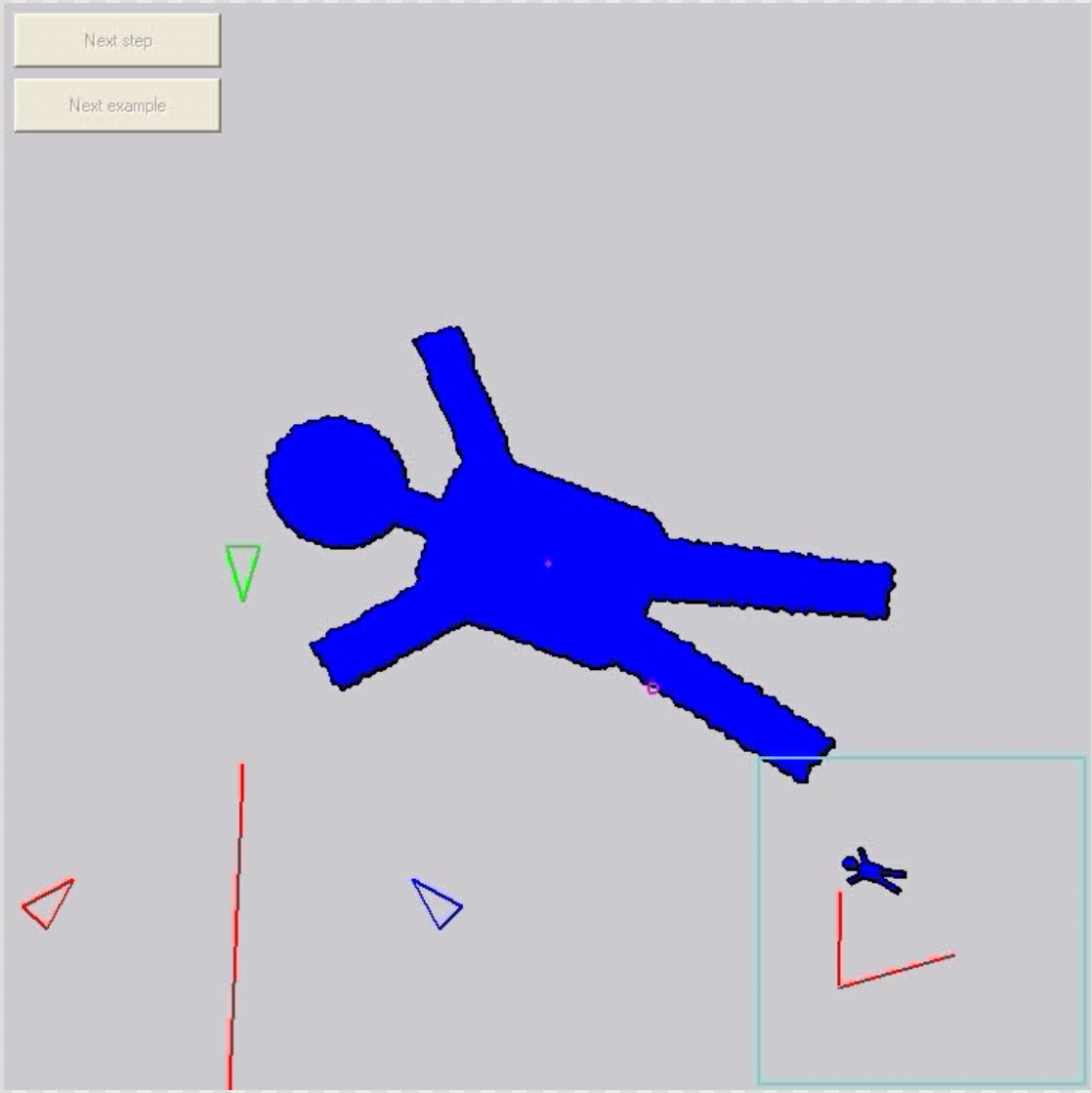
- Base de datos de 24 objetos:
  - 19 figuras geométricas sin agarres preestablecidos.
  - 5 herramientas con agarres relativos a la operación a efectuar.



Resultados experimentales (VII)

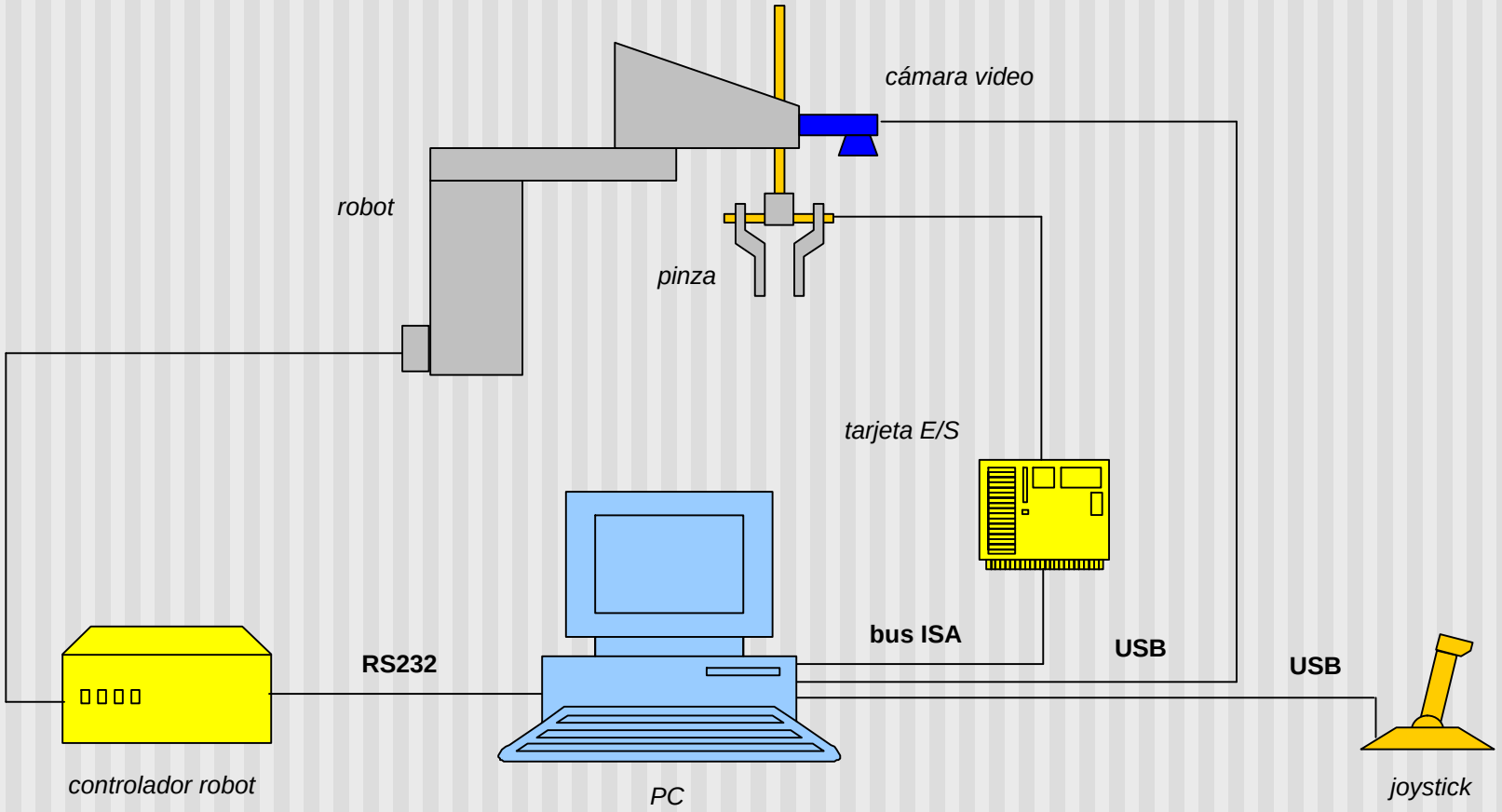


Resultados experimentales (VIII)



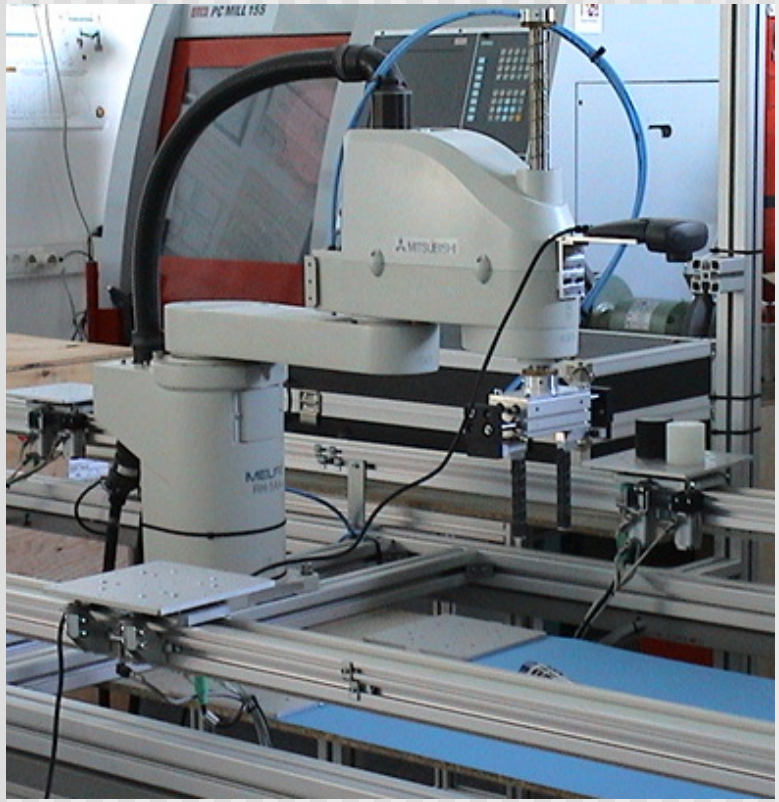
Resultados experimentales (IX)

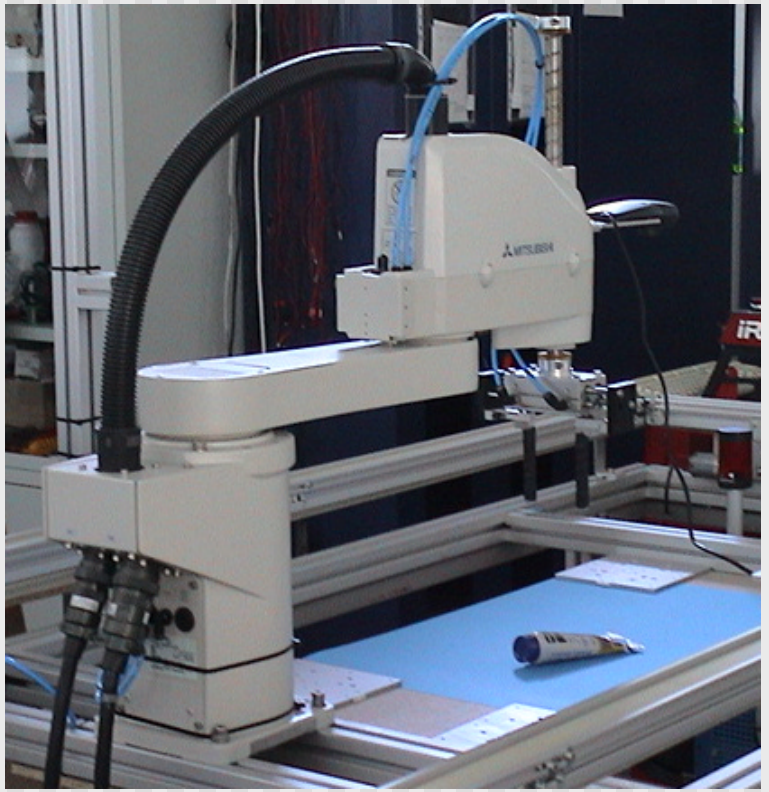
- Entorno real:
  - Robot SCARA con pinzas de dos y tres dedos.
  - Cámara cenital de bajo coste.



Resultados experimentales (X)

- Simplificación de la extracción del contorno:
  - Iluminación no controlada, pero fondo uniforme.
  - Extracción contorno mediante crecimiento de regiones y filtrado morfológico.





Resultados experimentales (XI)

video

Bloque VI

EJEMPLO DE APLICACIÓN