

# PROGRAMACIÓN DE ROBOTS CON LENGUAJE NATURAL

Ramón P. Neco, José M. Azorín, Nicolás M. García, José M. Sabater, Roque J. Saltarén  
*Departamento de Ingeniería*  
*División de Ingeniería de Sistemas y Automática,*  
*Escuela Politécnica Superior de Elche*  
Universidad Miguel Hernández  
03202 – Elche (Alicante), Spain  
e-mail: [ramon.neco@umh.es](mailto:ramon.neco@umh.es)

## Resumen

*En este artículo se presenta una arquitectura de programación de un brazo robot de 7 g.d.l. utilizando reconocimiento de voz y lenguaje natural. El sistema completo incluye un módulo de aprendizaje de tareas nuevas y otro módulo de visión 3D para reconocimiento de las características del entorno. El módulo de lenguaje natural traduce los comandos del operador a una representación en un lenguaje intermedio para, que posteriormente se ejecute el programa mediante un intérprete.*

**Palabras clave:** Teleoperación, robótica, programación, reconocimiento de voz, procesamiento del lenguaje natural.

## 1. Introducción

El uso intensivo de sistemas robotizados en áreas de servicio y mantenimiento hará necesario el diseño de interfaces entre el operador y el robot que sean fáciles de utilizar. En este sentido, el uso del lenguaje natural puede ser un mecanismo eficiente y flexible para teleoperar un robot. Algunos estudios concluyen que para tareas “de bajo nivel”, como puede ser una tarea simple de guiado de un robot, siguen siendo más eficientes las interfaces tradicionales de teleoperación (brazo maestro o joystick). El uso del lenguaje natural por voz para la programación de robots es eficiente especialmente en tareas “de alto nivel”, es decir, aquéllas en las que un único comando expresado mediante una frase se puede traducir a una acción compleja que incluye varias tareas individuales [3].

Algunos autores han realizado estudios sobre la aplicación de las técnicas de procesamiento de lenguaje natural a la programación de robots. El primer sistema completo fue el robot SAM (*Speech*

*Activated Manipulator*), desarrollado en los laboratorios Bell de AT&T [1]. Este robot se comunicaba con el operador humano por medio de lenguaje natural escrito o hablado por vía telefónica, e incluía multitud de sensores que le informaban sobre el contexto.

En un trabajo previo [6], se diseñó un lenguaje para la asistencia de un robot teleoperado, que se utilizaba como complemento en un sistema maestro-esclavo en dos sentidos: para especificar de forma sencilla restricciones de movimiento del robot y para “enseñar” al robot tareas elementales, que luego serán repetidas de forma autónoma.

En otros sistemas para robots móviles como los descritos en [2], [4], [7] y [8], se han desarrollado interfaces en lenguaje natural para robots móviles que se mueven en oficinas, en los que además de poder dar comandos y pedir información sobre los planes del robot, el usuario puede asociar nombres arbitrarios a localizaciones específicas del entorno.

En este artículo se presenta la arquitectura de un sistema de lenguaje natural para teleoperar un brazo robot, que realizará diversas tareas de servicio y ensamblaje. El sistema incluye un módulo de reconocimiento de voz continua en tiempo real que se comunica a través de una red local con el controlador del robot. La interfaz de lenguaje natural está diseñada de manera que sea flexible para los distintos módulos del sistema completo, que incluye: un módulo de aprendizaje de nuevas tareas y otro módulo de visión 3D para reconocimiento de las características del entorno.

## 2. Definición de la tarea

Existen numerosas aplicaciones industriales del reconocimiento del habla. Todas estas aplicaciones tienen algunas características comunes:

- La *dificultad de la tarea*, debida al tamaño y tipo de los vocabularios y a la dependencia del hablante que usa el sistema.
- La necesidad de *aprendizaje automático* por parte del sistema, lo cual supone un gran coste computacional por un lado y la necesidad de obtener un buen número de muestras de entrenamiento.
- La necesidad de un grado mínimo de *efectividad*.
- La existencia de *entornos de trabajo ruidosos*.

El sistema descrito en este artículo se ha aplicado a tareas de montaje. Se ha utilizado el brazo robot Mitsubishi PA-10, de 7 grados de libertad (figura 3), con una pinza simple en su extremo y montado sobre una plataforma fija. El entorno del robot está formado por una mesa y una colección de piezas, herramientas y objetos de diferentes tamaños y formas. El robot es instruido para que realice tareas de montaje utilizando estos elementos.

### 3. Estructura del sistema de reconocimiento del habla

En la figura 1 se muestra la estructura general de un sistema de reconocimiento del habla continua típico, dentro del marco del reconocimiento de formas.

Los pasos básicos del procesamiento de la señal son los siguientes:

1. *Análisis de características*. Se realiza un análisis frecuencial y/o temporal de la señal de voz para obtener vectores de observación o de características.
2. *Sistema de reconocimiento de unidades*. Se debe realizar la elección de la *unidad* de reconocimiento adecuada para nuestra aplicación. En nuestro sistema se ha elegido como unidad la palabra, debido a que nuestro vocabulario no es demasiado grande y la tarea a la que se aplica el reconocimiento es restringida. Las palabras se modelan usando modelos ocultos de Markov [9].
3. *Decodificación léxica*. En esta fase se añaden restricciones al sistema de reconocimiento de unidades de tal forma que las secuencias de unidades consideradas correspondan a secuencias de unidades de voz que existan en un diccionario de palabras.

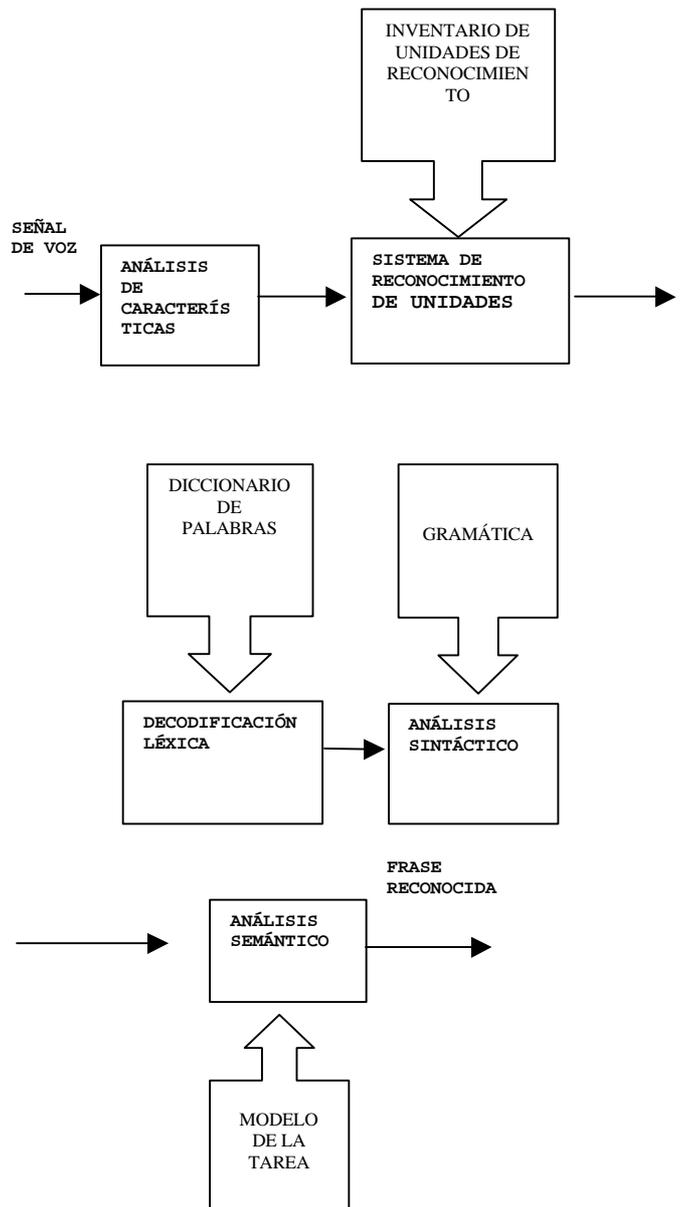


Figura 1: Estructura global de un sistema de reconocimiento del habla continua

4. *Análisis sintáctico*. En esta fase se añaden restricciones al sistema de reconocimiento de unidades de tal forma que las secuencias de unidades consideradas se correspondan con secuencias de unidades que formen palabras (que ya se consiguió en la decodificación léxica) y que estas palabras formen una secuencia válida según se especifica en una *gramática de palabras*.
5. *Análisis semántico*. Esta fase añade nuevas restricciones a las posibles secuencias de reconocimiento.

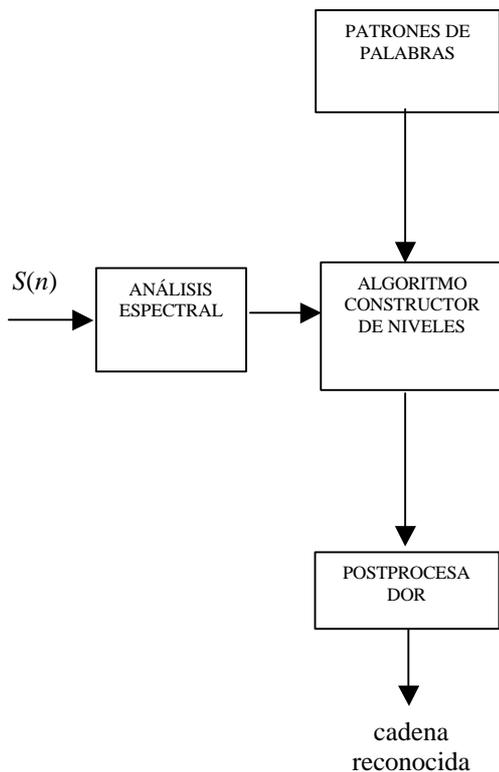


Figura 2: sistema típico de reconocimiento de palabras conectadas usando modelos de Markov

La metodología utilizada en nuestro sistema de reconocimiento del habla continua es la basada en los *modelos ocultos de Markov (Hidden Markov Models, HMM)*. [9] En la figura 2 se muestra un sistema típico de reconocimiento de palabras conectadas usando HMMs.

Los pasos que se siguen son los siguientes:

1. *Análisis espectral*. La señal de voz  $s(n)$  se convierte en un conjunto de vectores LPC [9] o en un conjunto de vectores de características. Esto define la secuencia de observación  $O$  de la cadena desconocida.
2. *Algoritmo de construcción de niveles*. Un nivel es una posición de palabra en una cadena. La secuencia de vectores de características de la cadena desconocida se compara con los HMMs de palabra única usando un algoritmo de puntuación de Viterbi [9]. La salida de este proceso es un conjunto de cadenas candidatas, generalmente de longitudes diferentes, ordenadas por puntuaciones.
3. *Postprocesador*. Sobre la cadena obtenida se realizan pruebas adicionales (por ejemplo, de duración), para eliminar candidatos no probables. El postprocesador elige la cadena más probable para las cadenas candidatas.

#### 4. Procesamiento del lenguaje natural

En esta sección se describe el esquema general de procesamiento del lenguaje natural usado en nuestra aplicación. Una vez que disponemos de la transcripción de la voz en un texto ASCII obtenida por el módulo de reconocimiento del habla, el módulo siguiente consiste en interpretar la frase y traducirla a un lenguaje comprensible por el robot.

Existen dos enfoques generales para realizar este procesamiento:

- Diseño
- Aprendizaje o adaptación

El *diseño* consiste en crear y programar una gramática específica para nuestra aplicación y realizar las tres fases típicas de análisis:

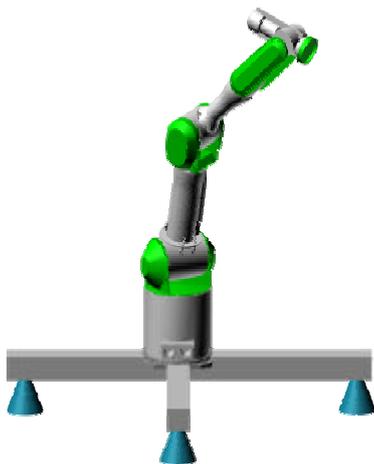


Figura 3: Brazo robot Mitsubishi PA-10 de 7 g.d.l.

1. Análisis *léxico*: consiste en identificar cada unidad mínima de la frase.
2. Análisis *sintáctico*: consiste en identificar la estructura sintáctica de la frase, obteniendo un árbol sintáctico.
3. Análisis *semántico*: consiste en interpretar y traducir la frase.

Mediante las técnicas de *aprendizaje* o *adaptación*, el sistema se adapta a las restricciones sintácticas y semánticas de nuestra aplicación a partir de ejemplos [5]. Pueden usarse, por ejemplo, redes neuronales recursivas tipo RAM (*Recursive Hetero-Associative Memories*), que básicamente consisten en un codificador que transforma la frase en una representación analógica interna y un decodificador que transforma esa representación interna en un orden o programa para el robot. El funcionamiento del codificador y decodificador es lo que se debe aprender.

El análisis léxico y análisis sintáctico son las fases más algorítmicas del proceso. Existen varios algoritmos que se pueden utilizar, independientemente de la aplicación. El objetivo de estas dos fases es obtener una estructura en árbol que represente la sintaxis de la orden de entrada.

El análisis semántico es dependiente de la aplicación. En esta fase queremos obtener el *significado* de la frase para poder traducirla al lenguaje objeto. Por tanto, es la parte más crítica y menos algorítmica del diseño. En la figura 4 se muestra el esquema de un sistema típico de procesamiento del lenguaje natural.

En las fases de análisis sintáctico y semántico se intenta obtener una estructura con significado a partir de la sucesión de palabras recibidas del reconocedor de voz. En el caso de la interfaz con un robot, esta estructura puede ser representada directamente por la secuencia de primitivas para el robot en el lenguaje destino.

En la mayoría de sistemas de procesamiento del lenguaje natural, y en particular en la interfaz con un robot, es necesario realizar dos fases posteriores al análisis semántico: la *integración del discurso* y el *análisis pragmático*.

- En la fase de *integración del discurso* se obtienen las coordenadas de los objetos y localizaciones indicadas por el operador. Estas coordenadas se obtendrán finalmente consultando la base de datos del sistema.
- En la fase de *análisis pragmático* se traducen las instrucciones intermedias a las que se ha llegado

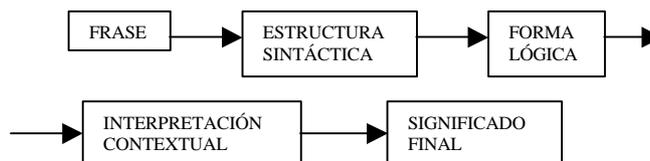


Figura 4: Esquema de un sistema de procesamiento del lenguaje natural.

a las instrucciones directamente entendibles por el robot (o las instrucciones definidas en el protocolo de comunicación con el robot).

## 5. Desarrollo del sistema de programación con lenguaje natural de un robot teleoperado

### 5.1 Arquitectura del sistema

El sistema se compone de tres módulos básicos que están interrelacionados: el módulo de lenguaje natural, el módulo de aprendizaje automático y el módulo de reconocimiento visual de la escena.

Estos tres módulos se comunican a través de la información almacenada en dos bases de datos: una base de datos de *tareas*, donde el sistema de aprendizaje almacena las instrucciones de las tareas aprendidas y una base de datos del entorno, donde el sistema de visión almacena la representación obtenida del entorno del robot. En la figura 5 se muestra la relación entre los tres módulos, donde L significa que el módulo accede a la base de datos para leer y E para escribir.

### 5.2 Módulo de procesamiento del lenguaje

La representación independiente del contexto que se ha elegido para la aplicación es un lenguaje intermedio que denominaremos *lenguaje destino*. Este lenguaje destino está formado por las primitivas básicas que puede ejecutar el robot, pero tienen un nivel de abstracción superior para que sea más sencillo el procesamiento del lenguaje. Los lenguajes de programación existentes para robots son inadecuados como lenguajes intermedios, debido principalmente a dos razones: (1) las funciones de movimiento y las representaciones del entorno obtenidas por los sensores típicos de un robot se parecen poco a las representaciones que un operador tiene en mente cuando emite un comando en lenguaje natural; (2) las estructuras de programación típicas de

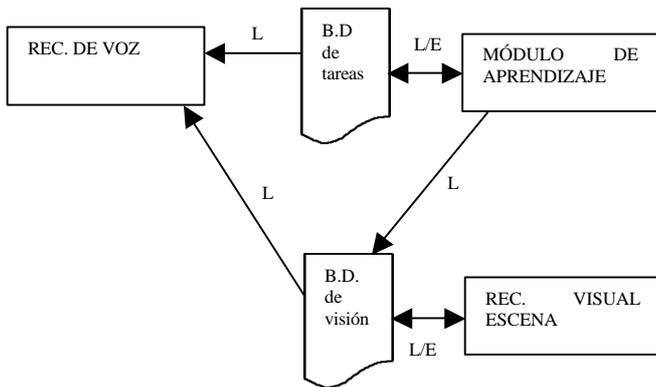


Figura 5: Arquitectura del sistema

estos lenguajes (secuencia, selección y repetición) son muy distintas a las estructuras que se obtienen después de analizar una frase en lenguaje natural.

El lenguaje intermedio usado en nuestro sistema se basa en procesos concurrentes, con la posibilidad de realimentación por voz y visión, con el objetivo de facilitar el proceso de traducción al lenguaje directamente ejecutable por el robot.

Algunas de las primitivas del lenguaje destino diseñado son las siguientes:

*move(punto)*  
mueve el extremo del brazo al punto *punto*

*moverD(dirección)*  
mueve el brazo del robot en la dirección especificada, hasta que se le da una orden de parada.

*stop(tiempo)*  
mantiene el brazo parado durante *tiempo* segundos

*approx(punto, distancia)*  
aproxima el extremo del brazo desde la posición actual hasta el *punto*, a una distancia *distancia* en centímetros. Si la distancia es negativa, se entiende que el operador desea alejarse del punto.

*open / close*  
abre o cierra la herramienta del extremo del brazo

*movePlane(plane)*  
mueve el extremo del robot al punto más cercano del plano especificado.

El módulo de reconocimiento de voz continua tiene como salida una lista de frases

posibles, ordenadas por probabilidad de ocurrencia. A continuación se realiza una fase de análisis morfológico en la que se analizan las concordancias entre palabras. Puesto que las unidades básicas del reconocimiento son las palabras, en la mayoría de los casos no es necesario este análisis morfológico.

En la siguiente etapa se realiza el análisis sintáctico y semántico conjunto, utilizando un tipo especial de las denominadas *redes de transición* [9], a partir de las cuales se obtiene la estructura sintáctica en forma de árbol.

En cada uno de los nodos de este árbol se especifican las instrucciones en lenguaje intermedio que deben ejecutarse. Un módulo de nuestro sistema llamado *intérprete de código intermedio* se encarga de ejecutar las acciones obtenidas del análisis realizando un recorrido del árbol en profundidad, de izquierda a derecha, aplicando las instrucciones a medida que se van encontrando.

En otros sistemas con interfaces de lenguaje natural se han utilizado otros modelos distintos para el lenguaje intermedio (como lenguajes de máquinas abstractas de registros, o lenguajes basados en modelos de sistemas de producción). La elección de nuestro modelo se basa en que se adapta muy bien a las estructuras y modelos que se obtienen después del análisis de una frase en lenguaje natural.

La aplicación contiene un módulo que permite adaptar la gramática diseñada a la tarea específica que se está desarrollando. Partiendo de una gramática base, este módulo permite modificar la gramática añadiendo nuevas reglas gramaticales y palabras al diccionario, proceso que se hace antes de comenzar la teleoperación. El sistema permite, además, introducir palabras nuevas desconocidas por el robot durante la teleoperación. Se ha desarrollado también un *compilador* de gramáticas que le permite al usuario diseñar su propia gramática y compilarla a un formato entendible por el reconocedor de voz y por el módulo de lenguaje.

El algoritmo de análisis utilizado consiste en recorrer las redes de transición [2], comenzando por la red principal, y leyendo secuencialmente la sentencia de entrada. En la figura 6 se muestra un ejemplo de red de transición. Cuando de un estado se pueda pasar a varios estados en la misma u otras redes, se elige uno de ellos siguiendo un orden preestablecido, guardando en una lista de estados de retorno el resto. Esta lista se utiliza para volver hacia atrás (*backtracking*) en la búsqueda de la secuencia de estados en el caso en que el camino seguido no nos lleve a interpretar la sentencia completa. De esta forma, se recorren los distintos estados hasta que se encuentra una secuencia de análisis de la sentencia de

entrada, la cual proporciona la base para su traducción al lenguaje destino. En las transiciones entre estados se hacen varias comprobaciones semánticas, de forma paralela al análisis sintáctico, tales como:

- Si la categoría gramatical correspondiente al estado tiene sentido en el lugar de la frase que aparece.
- Si existe concordancia morfológica (género, número, etc.). Si ocurre algún error en esta fase, pero se puede llegar a una secuencia de órdenes en el lenguaje destino del robot, se ejecuta la sentencia dando un aviso al operador.
- Comprobaciones de contexto: por ejemplo, si la palabra nombrada tiene sentido semántico dado el estado actual del robot y de su entorno. Si el operador da una orden como “Deja el tornillo en la caja”, la comprobación semántica de contexto consistirá en comprobar si efectivamente el robot tiene un tornillo en su pinza.

Una vez obtenida la estructura del comando emitido por el operador, el intérprete de código intermedio es el encargado de realizar la ejecución directa en el robot, como se ha explicado más arriba. El intérprete también se encarga de transformar las referencias a los objetos que se hacen en el comando en lenguaje natural en números reales que representan las coordenadas de los objetos en el espacio tridimensional. Estos valores se obtienen accediendo a la base de datos de visión mostrada en el diagrama de la figura 5.

La gramática utilizada por el sistema es leída en cada sesión de trabajo desde un fichero de texto. En este fichero se representan las reglas gramaticales utilizando la notación de las gramáticas independientes del contexto, junto con las acciones en el lenguaje intermedio que debe ejecutar el robot. El sistema interpreta este fichero y lo transforma en un fichero compilado (*gramática compilada*), que es directamente utilizable por los sistemas de reconocimiento de voz y de lenguaje natural

Para que el robot sea capaz de interpretar y ejecutar un comando en lenguaje natural que contiene preposiciones espaciales, la representación geométrica del entorno debe incluir información semántica. Por ejemplo, en comandos como “Coge la pieza cuadrada y colócala en el otro extremo de la mesa”, el sistema debe interpretar la expresión espacial “en el otro extremo” de manera adecuada en el estado actual del entorno del robot. Otro ejemplo: en un comando como “Coloca la pieza junto a la herramienta”, el sistema debe interpretar correctamente la expresión espacial “junto a”.

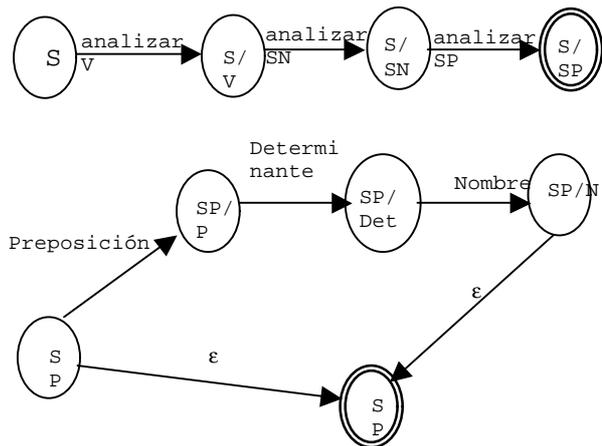


Figura 6: Red de transición de estados

Para interpretar estas expresiones espaciales, la representación geométrica debe enriquecerse con un conjunto de restricciones. Estos dos elementos (geometría más restricciones) caracteriza nuestro modelo sobre el entorno del robot, que permite representar nuestra *intuición* sobre los objetos a los que se hace referencia en los comandos en lenguaje natural.

En la tabla 1 se muestra un subconjunto muy reducido de reglas de una gramática que puede usarse en el sistema. Este subconjunto de reglas representa la estructura de comandos para ordenar al robot que obtenga objetos con su pinza. Junto a cada regla aparece la regla semántica asociada para obtener la traducción al lenguaje intermedio del robot.

REGLA SINTÁCTICA	REGLA DE TRADUCCIÓN
F -> V OB	$T(F) = T(V)(T(OB))$
OB -> Nombre1 Expr_espacial Nombre2	$T(OB) = T(Expr\_espacial)(T(Nombre1), T(Nombre2))$
OB -> Nombre1	$T(OB) = T(Nombre1)$
Expr_Espacial -> "que hay junto a"	$T(Expr\_Espacial) = f_{JUNTO\_A}(C_1, C_2)$
Nombre -> Det N	$T(Nombre) = T(N)$
Det -> "el"	
Det -> "la"	
N -> "estante"	$T(N) = E$
N -> "pieza"	$T(N) = P$
N -> "coge"	$T(N) = f_{COGER}(elemento)$

Tabla 1: Subconjunto de reglas de la gramática

En la primera columna de la tabla aparecen las reglas sintácticas, las cuales capturan la estructura sintáctica que pueden tener los comandos emitidos por el operador. Se han incluido sólo las reglas adecuadas para representar el comando “*coge la pieza que hay junto al estante*”. En las reglas sintácticas se ha escrito, en la parte izquierda del símbolo “->” el nombre de la categoría que representa esa regla, y en la parte derecha una secuencia de símbolos que representa la estructura que puede tener esa categoría. Por ejemplo, la primera regla  $F \rightarrow V OB$ , expresa que una frase está compuesta por un verbo seguido de un objeto. La segunda regla,  $OB \rightarrow Nombre1 Expr\_espacial Nombre2$ , expresa que un objeto puede estar compuesto por un nombre, una expresión espacial y otro nombre, como en la expresión “*la pieza que hay junto al estante*”, donde la expresión espacial es “*que hay junto a*”. La tercera regla,  $OB \rightarrow Nombre1$ , expresa que un objeto también puede expresarse como un nombre. El resto de reglas sintácticas se interpretan de forma análoga. Las palabras literales que puede emitir el operador se han representado en cursiva en la parte derecha de las reglas sintácticas. Por ejemplo, la regla  $N \rightarrow$  “*estante*” indica que el operador puede emitir la palabra literal “*estante*”, que el reconocedor lo interpretará como un nombre.

La traducción a lenguaje intermedio del comando “*coge la pieza que hay junto al estante*” es:

$$f_{COGER}(f_{JUNTO\_A}(P, E))$$

donde  $f_{COGER}$  representa la función del robot para coger un objeto. Esta función tiene un parámetro que es el objeto a coger. La función  $f_{JUNTO\_A}$  es una función de acceso a la base de datos de visión, que toma dos parámetros  $P$  y  $E$ , que son dos conjuntos de objetos, y devuelve una lista con todos aquellos objetos del primer conjunto ( $P$ ) que cumplen la relación geométrica “*estar junto a*” respecto a los elementos del segundo conjunto ( $E$ ). En nuestro ejemplo,  $P$  representa el conjunto de “*piezas*” y  $E$  el conjunto de “*estantes*” que existan en el entorno del robot.

La lista de objetos que obtiene como resultado la función  $f_{JUNTO\_A}$  es pasada como parámetro de entrada a la función  $f_{COGER}$ . Si esta lista contiene más de un objeto, se produce un problema de ambigüedad en el comando del operador ya que el sistema habrá detectado más de una pieza que esté junto al estante. En ese caso la función  $f_{COGER}$  devuelve un código de error y el sistema interactúa con el operador hasta obtener el identificador de un único objeto.

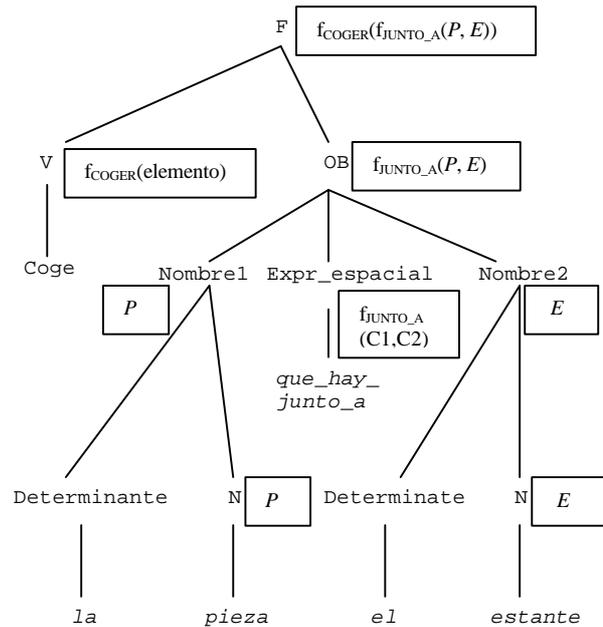


Figura 7: Estructura global de un sistema de reconocimiento del habla continua

Para poder obtener esta traducción, se necesitan las reglas que se muestran en la segunda columna de la tabla. Estas reglas las utiliza el sistema de lenguaje natural para traducir al lenguaje intermedio cada una de las expresiones. Por ejemplo, la primera regla de traducción,  $T(F) = T(V)(T(OB))$ , expresa que la traducción de la frase a lenguaje intermedio está formada por la función del robot correspondiente al verbo de la frase pasándole como parámetro la traducción correspondiente al objeto (el símbolo  $T$  significa traducción). En nuestro ejemplo,  $T(V) = f_{COGER}$  y  $T(OB) = f_{JUNTO\_A}(P, E)$ , siendo la traducción completa del comando  $f_{COGER}(f_{JUNTO\_A}(P, E))$ .

El resto de reglas de traducción representan las traducciones parciales para obtener la traducción completa del comando. Por ejemplo, la regla de traducción para la regla sintáctica  $N \rightarrow$  “*pieza*” es  $T(N) = P$ , donde  $P$  es el conjunto de los objetos del entorno que se consideran “*piezas*”. Esta regla de traducción implica un acceso a la base de datos del entorno para obtener una lista con los identificadores de todos los objetos que sean “*piezas*”.

En la figura 7 se muestra un árbol sintáctico para el análisis del comando “*coge la pieza que hay junto al estante*”. Junto a cada nodo aparece un cuadro con la traducción parcial obtenida, siendo la traducción que aparece junto a la raíz del árbol la traducción completa del comando.

## 6. Conclusiones y trabajo futuro

Se ha diseñado un sistema teleoperado que admite órdenes complejas usando un reconocedor de voz continua, con la estructura apropiada para su integración con un módulo de aprendizaje y un módulo de visión 3D. La principal conclusión que obtenemos es que para el diseño de interfaces en lenguaje natural para robots, es conveniente el uso de un lenguaje objeto adecuado para las estructuras obtenidas después del análisis del comando de entrada.

El trabajo futuro incluye: optimización de los algoritmos de análisis sintáctico y semántico y el diseño de una representación geométrica del entorno que sea general y que sea adecuada para la representación de los objetos en la base de datos para resolver las referencias semánticas.

### Agradecimientos

Este trabajo ha sido financiado por la Comisión Interministerial de Ciencia y Tecnología (CICYT) a través del proyecto TAP98-1083-C02-01.

### Referencias

- [1] Brown, M.K., Buntschuh, B.M., Wilpon, J.G., "SAM: A Perceptive Spoken Language Understanding Robot", *IEEE Trans. on Systems, Man and Cybernetics*, 22, November/December 1992.
- [2] Crangle Collen, Suppes Patrick (1994) *Language and learning for robots*, Center for the Study of Language and Information (CSLI) Lecture Notes, N. 41.
- [3] Ferre, M., Macías-Guarasa, J., Aracil, R., Barrientos, A. "Voice Command Generation for Teleoperated Robot Systems", *7<sup>th</sup> IEEE International Workshop on Robot and Human Communication 1998 (RO-MAN '98)* pp. 679-685, vol. 2, ISBN 4-921073-00-7. Takamatsu, Kagawa (Japan), September 1998.
- [4] Laengle T., Lueth, T.C. "KANTRA: A Natural Language Interface for Intelligent Robots", *Proc. of the 4<sup>th</sup> International Conference on Intelligent Autonomous Systems*, 1995.
- [5] Salerno, J.J. "An application of a recurrent neural model for parsing natural language", *International Journal on Artificial Intelligence Tools [Architectures, Languages, Algorithms]*, 6 Iss:3 p. 397 – 419. Sept 1997.
- [6] Sato Tomomasa, Hirai Shigeoki (1987) "Language-Aided Robotic Teleoperation System (LARTS) for Advanced Teleoperation". *IEEE International Conference on Systems, Man and Cybernetics*.
- [7] Torrance, M.C. *Natural Communications with Robots*, Master's thesis, MIT, Department of Electrical Engineering and Computer Science, Cambridge, MA, 1994.
- [8] Yared, W.I., Sheridan, T.B., "Plan Recognition and Generalization in Command Languages with Application to Telerobotics", *IEEE Trans. on Systems, Man and Cybernetics*, vol 21 n°2, 1991.
- [9] Wang, X., Pols, L. "A preliminary study about robust speech recognition for a robotics application", Institute of Phonetic Sciences, University of Amsterdam, *Proceedings 21* (1997), 11—20.