AGENTES ROBÓTICOS Y ENTRETENIMIENTO

RESUMEN TRABAJO FIN DE GRADO - JULIO 2013

Introducción

La robótica es una de las múltiples ramas de la tecnología dedicada al diseño, fabricación y programación de robots. Dentro de la robótica se coordinan varias especialidades como la informática, la mecánica, la electrónica o la inteligencia artificial.

Este trabajo se centra en el apartado de la informática y, en concreto, en la programación con el fin de obtener una interacción entre robots.

Un robot es un sistema electromecánico controlado y programado que puede llegar a moverse o a realizar diversas acciones en función del ambiente que le rodea. En la actualidad, los robots se utilizan en diversos ámbitos como: Industrial, medicina, tareas repetitivas, doméstico, entretenimiento, etc.

La definición del término robot abarca muchos tipos, por lo que existen diversas maneras de clasificarlos. Una de ellas, en función de su arquitectura, se detalla a continuación ya que es la que incluye a los robots humanoides, que han sido los utilizados para la realización de este trabajo.

Por lo tanto, en función de su arquitectura se distinguen cinco tipos:

- **Robots poliarticulados** – Robots que en su mayoría están fijos y que pueden mover sus terminales dentro de un determinado espacio de trabajo para realizar diversas acciones en función de sistemas de coordenadas.

- **Robots móviles** – Robots capaces de moverse mediante sistemas locomotores gracias a información obtenida por sensores.

- **Robots humanoides** – Robots que tratan de reproducir en parte o totalmente la cinemática humana para ser capaces de realizar diferentes movimientos.

- **Robots zoomórficos** – Robots que tratan de reproducir los sistemas locomotores de varios seres vivos y que a su vez se dividen en no caminadores y caminadores.

- **Robots híbridos** – Robots que combinan dos o más de las estructuras nombradas anteriormente.

Dentro de las anteriores clases, este trabajo se centra en el uso de robots humanoides. Estos robots son aquellos encargados de reproducir total o parcialmente determinadas funciones humanas. A menudo este tipo de robots imitan al ser humano en cuanto a apariencia y movimientos y, actualmente, se encuentran en continuo desarrollo e investigación. Una de sus diversas utilidades es la del entretenimiento, que es en la que se basa este trabajo.

Dentro de los androides o robots humanoides se encuentran los robots NAO, desarrollados y fabricados por la empresa francesa *Aldebaran Robotics*, modelo que se ha utilizado en la realización de este trabajo.

Junto a estos robots, se proporciona la herramienta informática *Choregraphe*, desde la cual se pueden controlar y gestionar las diferentes funciones del robot de manera sencilla.

Además, *Choregraphe* ofrece la posibilidad de programar en distintos lenguajes para dotar al robot de nuevas habilidades creadas desde cero o a partir de las que ya ofrece el programa. En el caso de este trabajo, se eligió *Python*.

Marco Regulador

Actualmente, cada vez son más los proyectos que implican el uso de robots y, en concreto, en ámbitos como entretenimiento o salud uno de los robots que destacan son los robots humanoides NAO, utilizados en este trabajo.

Uno de éstos, es el proyecto europeo llamado FEELIX que se está desarrollando en colaboración entre ocho universidades y empresas de robótica de la Unión Europea. Uno de los objetivos del mismo es mejorar aún más la iteración que tiene NAO con las personas utilizando señales no verbales.

De momento, en este proyecto se ha conseguido diseñar a NAO para imitar las emociones de un niño de un año y que sea capaz de crear vínculos con las personas que lo tratan amablemente. También puede detectar emociones humanas mediante una serie de claves como el lenguaje corporal y las expresiones faciales. Además, es capaz de recordar iteraciones con personas y memorizar rostros.

Actualmente, la última versión del robot NAO es la llamada ASK NAO (Autism Solution for Kids), creada también por Alderaban Robotics y que se empezará a usar para ayudar a niños autistas. En una primera fase ha sido probado en un colegio del Reino Unido.

El robot viene con juegos y aplicaciones diseñadas por médicos especialistas para fomentar la atención de los niños autistas y ayudarles a mejorar su estado. Al igual que los robots NAO utilizados en este trabajo, ASK NAO va equipado con sensores, dos cámaras, cuatro micrófonos, sintetizador de voz, dos altavoces y luces LED.

Objetivos

El objetivo de este trabajo era conseguir construir un sistema que fuera capaz de proporcionar entretenimiento, en forma de interpretación de obras de teatro, mediante el uso de agentes robóticos, en concreto robots NAO.

Para ello se utilizará un archivo de texto XML, en el que estarán incluidas las frases y los tipos de movimiento que deben realizar los robots. Éstos, mediante un *script* en *Python*, se coordinarán para ofrecer una pequeña representación basada en el contenido del archivo XML, que se le pasará al *script* como parámetro de entrada.

Este sistema debía tener las siguientes características:

- El entretenimiento debía consistir en voz y movimientos y estar estructurado dentro de un archivo de texto, proporcionado como entrada al sistema, en el que se indicase como serían las intervenciones de cada uno de los robots.

- Estas intervenciones podían ser: decir una frase, realizar un movimiento o ambos a la vez.

- La aplicación debía servir para cualquier archivo de texto que se le pasase siempre que estuviera correctamente estructurado.

- Los movimientos deberían programarse desde cero, algunos serían básicos y otros un poco más específicos en función del tipo de diálogo.

- En las piezas, puede intervenir solo un robot o dos robots. En este último caso, había que conseguir coordinar los robots de manera que se tuviese la sensación de estar ante una conversación normal, es decir, que no se solapasen con las frases o movimientos ni estuviesen demasiado tiempo quietos.

Para conseguir todo esto, el código utilizado debía ser único para ambos robots y servir para cualquier archivo XML creado a partir de las pautas necesarias.

Solución Técnica

La realización de este trabajo se dividió en tres bloques complementarios, consiguiendo que:

Estos tres bloques se hicieron primero con diálogos de prueba y, cuando se comprobó que funcionaban correctamente, se escogieron los diálogos de las demos finales y se programaron todos los movimientos que usarían para las últimas pruebas.

En la primera fase del trabajo, el objetivo era que un robot hablase. Para ello, debía leer un fichero de entrada XML y decir las frases contenidas en el atributo "text" de las etiquetas del mismo. Para esto utiliza el módulo *"ALTextToSpeech"* y, en concreto, la función *"say(string toSay)"*. Mediante el uso de esta función el robot es capaz de decir las frases que se indiquen en el parámetro *"toSay".*

En la segunda fase, el objetivo era que un robot fuese capaz de realizar una serie de movimientos deseados. Estos movimientos se grabaron utilizando *Choregraphe* y los robots reales y después se exportaron como código como funciones. Estas funciones utilizan el módulo "ALMotion" y, en concreto, la función *"walkTo(float x, float y, float theta)"* en los casos en los que el robot anda y la función *"angleInterpolationBezier(names, time, keys)"* en el resto de movimientos. Para indicar qué movimiento deben realizar los robots, se añadió el atributo "move" a las etiquetas del archivo XML de entrada.

En la tercera fase, el objetivo era coordinar dos robots para que interpretasen un diálogo contenido el archivo XML de entrada. Para ello, se añadió el atributo "name" a las etiquetas del archivo XML, que indica qué robot debe intervenir. Además, el código se envía a un único robot que será el coordinador de la conversación. Este robot, cada vez que lee una etiqueta, comprueba el atributo "name" y cambia los valores de *"ip"* y *"port"* por los valores del robot al que le corresponda intervenir.

A continuación se describe la estructura que deberá tener un fichero de entrada XML para que una obra sea interpretada correctamente por uno o dos robots.

Este archivo XML deberá estar en el mismo directorio en el que se encuentre el *script* final *"teatroNAO.py"* y en él se podrán escribir las etiquetas deseadas, teniendo en cuenta que los contenedores de las etiquetas deberán ser *<action></action>.* Además, dentro de cada etiqueta podrán tomar valores los siguientes atributos:

"name" – Podrá ser *"robot1", "robot2"* o *"robot1androbot2"* e indicará qué robot será el que realizará las acciones definidas en esa etiqueta. Respectivamente, éstas podrán ser realizadas por sólo uno de los robots o por ambos.

En caso de que ***name="robot1"*** *o* ***name="robot2"****,* se podrán incluir los siguientes atributos:

"text" – Incluirá la frase que se desea que diga el robot indicado en el atributo *"name"*.

"move" – Contendrá el tipo de movimiento que se quiera que realice el robot indicado en el atributo *"name".* Los valores que puede tomar este atributo se muestran en la tabla siguiente:

GESTO DEL ROBOT	VALORES DEL ATRIBUTO "move"
Movimiento propio de la acción de cada uno de los verbos	*"hablar", "negar", "afirmar", "preguntar", "enfadar", "saludar", "gritar"*
Señalan de seis formas distintas	*"point1", "point2", "point3", "point4", "point5", "point6"*
Hablan y señalan de seis formas distintas	*"hablarpoint1", "hablarpoint2", "hablarpoint3", "hablarpoint4", "hablarpoint5", "hablarpoint6"*
Niegan con la cabeza y con ambos brazos	*"negarBrazos"*
Realizan una reverencia	*"reverencia"*
Hacen un gesto con el brazo animando a que se les siga	*"venir".*
Mueven un brazo haciendo el gesto de mirar la hora	*"reloj"*
Se ponen rectos y se llevan la mano al pecho	*"pecho"*
Se llevan la mano al pecho para indicar que tienen frío	*"frio"*
Mueven un brazo y se señalan los ojos o una oreja	*"ver"*
Caminan sin hablar, por lo que no tiene validez el atributo *"text"* y se hace imprescindible incluir el atributo *"distance"* que indica en metros la distancia que se desea que camine el robot	*"andar"*
Caminan hablando de dos maneras distintas, por lo que se hace imprescindible incluir el atributo *"distance"* que indica en metros la distancia que se desea que camine el robot	*"andarhablar1", "andarhablar2"*
Giran 180º y 90º hacia izquierda y derecha	*"giro1", "giro2", "giro3"*
Giran su cabeza a izquierda o derecha	*"giroCabeza1", "giroCabeza2"*
No realizan ningún movimiento	*"null"*

En esta tabla, los movimientos correspondientes a distintas formas de caminar (*move="andar", move="andarhablar1"* y *move="andarhablar2"*) y a diferentes formas de girar (*move="giro1", move="giro2"* y *move="giro3*") están basados en el uso de la función *"walkTo(float x, float y, float theta)".*

El resto de movimientos han sido programados desde cero para la realización de este trabajo.

En esta tabla se observa, en la última fila, que el valor del atributo *"move"* puede ser *move="null*". Esto será necesario si se desea que el robot hable sin realizar ningún movimiento y es imprescindible indicarlo siempre, ya que si una etiqueta no tiene atributo "move", el código ignorará su contenido.

"distance" – Indicará la distancia que tiene que caminar el robot, en metros. Recordar que este atributo únicamente es válido si *move="andar", move="andarhablar1"* o *move="andarhablar2".*

En caso de que ***name="robot1androbot2"***, se podrán incluir los siguientes atributos:

"move1" *y* ***"move2"*** – Contendrán el tipo de movimiento que se quiere que realice cada uno de los robots. Las parejas de valores que pueden tomar estos atributos son los siguientes:

- Los robots caminan sin hablar, por lo que no tiene validez el atributo *"text"* y se hace imprescindible incluir el atributo *"distance1"* si *move1="andar"* o *"distance2"* si *move2="andar"*, que indica en metros la distancia que se desea que camine el robot: *"andar"* y *"andar".*

- Los robots caminan hablando de dos maneras distintas, por lo que se hace imprescindible incluir el atributo *"distance1"* si *move1="andarhabar1"* o *"distance2"* si *move2="andarhablar2",* que indica en metros la distancia que se desea que camine el robot: *"andarhablar1"* y *"andarhablar2".*

- Los robots giran, cada uno, 90º hacia lados contrarios: *"giro2"* y *"giro3".*

"text1" y ***"text2"*** – Incluirán las frases que se desea que digan cada uno de los robots.

"distance1" – Indicará la distancia que tiene que caminar el robot, en metros. Recordar que este atributo únicamente es válido si *move1="andar"* o *move1="andarhablar1".*

"distance2" – Indicará la distancia que tiene que caminar el robot, en metros. Recordar que este atributo únicamente es válido si *move2="andar"* o *move2="andarhablar2".*

Conclusiones

Teniendo en cuenta los objetivos que se habían establecido al inicio de este trabajo, se pueden analizar los resultados obtenidos evaluando hasta qué punto se han logrado los mismos.

Conociendo el objetivo principal establecido, se puede comprobar su resultado final a través de varias pruebas.

Para ello, se recordará que había que lograr que dos robots NAO fuesen capaces de mantener una conversación, creada en un archivo externo, incluyendo diversos movimientos.

La versión final de este trabajo, consiste en un único *script* escrito en *Python* llamado *"teatroNAO.py"*. Este *script* es capaz de recibir como parámetro de entrada cualquier archivo XML y, si sigue la estructura adecuada, reproducir el diálogo que contenga en dos robots NAO.

Además, se puede ejecutar en cualquier ordenador que disponga de los elementos necesarios (*Python*, NAOqi y opcionalmente *Choregraphe*) y las pruebas pueden realizarse tanto en los robots reales como en un simulador, si se dispone de *Choregraphe*.

Resultados

A lo largo del desarrollo de este trabajo se han realizado numerosas pruebas para comprobar el correcto funcionamiento del código.

En su mayor parte, estas pruebas se realizaron en el simulador por no disponer físicamente de los robots reales en todo momento. No obstante, aproximadamente una o dos veces a la semana se ha trabajado con los robots reales para comprobar que todo avanzaba según lo previsto.

Finalmente, se crearon tres archivos XML diferentes que utilizan algunos de los movimientos previamente programados y que se utilizarían en las "*demos"* finales. Cada uno de estos archivos XML corresponde a un diálogo distinto y su resultado, al ejecutarse cada uno de ellos como parámetro de entrada en el *script "teatroNAO.py"*, es el siguiente:

Demo 1 – Utilizando el archivo *"obra1.xml"*, se puede comprobar la representación hecha por los robots NAO en el siguiente video:

`http://www.youtube.com/watch?v=cJu1VweV3BQ`

Esta Demo corresponde al Acto Primero, Escena primera, de la obra *"Hamlet"* de William Shakespeare.

Demo 2 – Utilizando el archivo *"obra2.xml"*, se puede comprobar la representación hecha por los robots NAO en el siguiente video:

`http://www.youtube.com/watch?v=E2pGcJWReEk`

Esta Demo corresponde al Acto Quinto, Escena primera, de la obra "Hamlet" de William Shakespeare.

Demo 3 – Utilizando el archivo *"obra3.xml"*, se puede comprobar la representación hecha por los robots NAO en el siguiente video:

`http://www.youtube.com/watch?v=S5Fa5zS3Q4c`

Esta Demo corresponde al Acto Segundo, Escena tercera, de la obra *"Macbeth"* de William Shakespeare.

El código creado para la realización de este trabajo ha resultado ser una herramienta de manejo sencillo y susceptible de ser modificada, ampliada y mejorada.

Se trata de una herramienta muy útil en el campo del entretenimiento, que puede ser utilizada como base en futuros desarrollos de más amplio contenido.

Además, ha servido para cumplir los objetivos principales del trabajo, obteniendo unos óptimos resultados en las pruebas realizadas.

Gracias al código del *script "teatroNAO.py"*, cualquier usuario puede crear sus propios diálogos mediante archivos XML, utilizando los movimientos disponibles y comprobar el resultado en los robots NAO reales.

En caso de no disponer de los robots reales, también es posible comprobar este resultado en el simulador ofrecido por *Choregraphe*.

Por lo tanto, este trabajo está enfocado principalmente a un uso académico y de investigación, ya que es complicado utilizarlo de manera cotidiana más allá de para simples simulaciones.

Aun así, en cuanto a investigación sí es algo importante ya que, como se ha mencionado antes, se puede ampliar y mejorar fácilmente por lo que puede ser utilizado como una base sencilla en la realización de otros proyectos más ambiciosos.

Líneas futuras

En caso de querer ampliar o modificar el código desarrollado en este trabajo, existen una serie de líneas futuras o trabajos de mejora de forma que éste proporcione más alternativas en cuanto a entretenimiento se refiere.

Entre estas posibles modificaciones, se podrían destacar las siguientes:

Añadir nuevos movimientos – El código desarrollado cuenta con una variada lista de movimientos, pero se podrían añadir muchos más en función de los diálogos que se quieran interpretar.

Añadir más robots – El trabajo está desarrollado para un diálogo entre dos robots, pero el código se podría mejorar para permitir interpretar escenas entre varios.

Obras más largas – Las demos finales son de corta duración, debido a que el objetivo era mostrar que el código funcionaba con diferentes tipos de archivos XML pasados como parámetro de entrada. Sin embargo, el código es capaz de interpretar diálogos de más duración.

Añadir melodías y LEDs – Las posibilidades de proporcionar entretenimiento por parte de los robots NAO son muy extensas y en este trabajo sólo se han utilizado una pequeña parte de ellas. Una mejora interesante sería crear nuevas animaciones, basadas en la reproducción de música o el uso de los LEDs de los que dispone NAO.

Generar obras dinámicamente – Los diálogos que interpretan los robots se pasan al código mediante un archivo XML como parámetro de entrada. Sin embargo, este código se podría utilizar como base para conseguir que los robots fuesen capaces de generar las obras dinámicamente.