

Framework para el modelado de Interacciones Humano-Robot basadas en aprendizaje no supervisado

Rafael Ramón Vigo, Luis Merino

Universidad Pablo de Olavide, Crta. Utrera km 1, Sevilla, rramvig, lmercab@upo.es

Fernando Caballero

Universidad de Sevilla, Crta. Camino de los Descubrimientos, s/n, Sevilla, fcaballero@us.es

Resumen

En el presente trabajo se propone un marco de referencia para el aprendizaje de interacciones entre personas y robots, basado en el uso conjunto de una técnica de aprendizaje sin supervisión y de un planificador de muestreo de configuraciones. Particularmente, se hace uso de los Modelos Mixtos Gausianos (GMMs en inglés) para modelar la interacción física de un robot y de una persona cuando este robot es teleoperado por alguien experto. De este modo, distintos comportamientos tales como evitar, acercarse o seguir a una persona pueden ser fácilmente derivados e incluso combinados gracias a las propiedades de las GMM. Los modelos aprendidos se integran en un planificador basado en muestreo, un RRT, bajo dos preceptos: primero, como una función de costes que permita ponderar el espacio de estados del robot como más afín a los comportamientos aprendidos y segundo, como sesgo del muestreo para descartar aquellas zonas menos probables según se haya concluido de las demostraciones. El algoritmo se ha probado exitosamente en el laboratorio usando un robot real y las trayectorias provistas por un experto.*

Palabras clave: Planificación óptima, Gaussian Mixture Models, Aprendizaje por demostración, Navegación Social.

1 Introducción

El proyecto europeo TERESA¹ considera la mejora de robots de telepresencia para que puedan navegar de forma autónoma y social en el interior de edificios. Para ello, este proyecto tiene en consideración el desarrollo de técnicas para la navegación eficiente y segura del robot mientras alcanza sus objetivos de navegación. Esta tarea supone un gran desafío cuando involucra a las personas que rodean o comparten el espacio con el robot y se consideran explícitamente en el modelado de la navegación.

La navegación social es un tarea compleja que se ha venido abordando bajo diferentes aproxima-

ciones en la comunidad robótica. Muchas de las técnicas actuales se basan en el aprendizaje automático de comportamientos socialmente aceptables a partir de datos reales recogidos en varias situaciones sociales, evitando la necesidad de programar dichos comportamientos de manera explícita. Así, los autores en [16] usan aprendizaje supervisado para inferir modelos de predicción de movimientos (trayectorias) humanas a partir de interacciones humano-robot en escenarios muy concurridos. En Luber et al. [11] se utiliza aprendizaje sin supervisión para caracterizar prototipos de navegación social que posteriormente son empleados para inferir un coste social sobre las trayectorias planificadas. En [7] se utiliza un modelo de fuerzas sociales en el que los parámetros de dicho modelo son aprendidos gracias al feedback dado por los usuarios.

Otra aproximación es aprender de las demostraciones, o aprendizaje por imitación [2]: un experto enseña a un robot cómo debería navegar entre personas. En el proyecto TERESA podemos sacar ventaja del hecho de que el robot es teleoperado por personas, de manera que se puede extraer información útil de estos usuarios. De este modo, usando como ejemplo dichos caminos o trayectorias, así como las configuraciones de rasgos más relevantes de la tarea ejecutada (como por ejemplo, la posición del robot respecto a las demás personas que lo rodean) abre la puerta a la extracción de las restricciones más relevantes que mejor puedan describir dichas trayectorias. La principal hipótesis que se plantea es que dichas trayectorias encierran las implicaciones sociales que las personas tienen en cuenta a la hora de realizar dichas tareas.

La aproximación de Aprendizaje por Refuerzo Invertido (Inverse Reinforcement Learning, IRL) [1] es una buena candidata para derivar dichos modelos: una función de recompensa (o de costes) es extraída a partir del comportamiento del experto y posteriormente se utiliza para obtener la correspondiente política que ha de seguir el robot. En [9], un planificador basado en IRL es empleado para aprender a partir de trayectorias de ejemplo que involucran interacción con personas, de manera que este aprendizaje incluya cierta con-

¹<http://teresaproject.eu/>

ciencia de los típicos comportamientos sociales que pueden darse en tales situaciones. IRL para navegación social también se utiliza en [13]. Sin embargo, mientras que en [9] los costes se emplean para planificar caminos, en [13] los autores emplean dichas técnicas para aprender políticas de navegación local, de manera que aprende el control directo sobre las acciones del robot.

Las anteriores aproximaciones de IRL hacen uso de los Procesos de Decisión de Markov (MDPs en inglés) como modelo subyacente del experto. Sin embargo, en general resulta complejo modelar problemas con MDPs debido a su complejidad computacional [12]. Nuestro objetivo es usar planificadores del estado del arte basados en muestreo, como el “optimal Rapidly exploring Random Trees (RRT*)”, para poder trabajar con espacios de configuraciones continuos. Estos planificadores ya razonan sobre los obstáculos presentes en el entorno, y el objetivo es incorporar en ellos información acerca de la tarea social llevada a cabo, extraída de los datos de ejemplo.

Los Gaussian Mixture Models (GMMs) ofrecen un marco de referencia flexible para modelar las relaciones entre los indicadores más relevantes que describen una tarea particular de navegación realizada por el robot. Los GMMs pueden extraer de manera no supervisada una buena representación de las distribuciones continuas de dichos indicadores, y ya ha sido probada su utilidad como modelos de representación de habilidades de robots en aplicaciones del paradigma de Programación por Demostración (PbD en inglés) [3].

A diferencia de [3], en este trabajo nuestro objetivo es utilizar los modelos GMMs extraídos de los datos conjuntamente con planificadores basados en el muestreo de estados, en particular “optimal Rapidly exploring Random Trees (RRT*)”. Este tipo de planificador intenta encontrar algún camino libre de obstáculos entre una configuración inicial y otra final, mientras optimiza una determinada función de costes. Para aprovechar las ventajas de ambos, nuestro objetivo es encontrar un camino seguro que imite un comportamiento determinado al permanecer dentro de los límites estadísticos del modelo obtenido. Para conseguir esto, se propone en primer lugar, sesgar el muestreo aleatorio de configuraciones hacia las regiones del espacio de configuraciones que mejor cumplan con el modelo de la tarea extraído de los datos; y, en segundo lugar, incluir una nueva función de costes en el RRT* para que pondere como más adecuadas aquellas configuraciones que sigan dichos comportamientos.

Una meta similar se persigue en [5]. En dicho trabajo, los autores también utilizan PbD, en el

que los modelos GMM se usan para extraer las restricciones estadísticas de una tarea concreta a partir de la realización de un número determinado de ejemplos, de manera similar a como se realiza en [4]. Después de esto, se utiliza un planificador RRT basado en el muestreo de configuraciones en el que se mantiene el árbol de búsqueda y expansión dentro de los límites estadísticos anteriormente modelados por las GMMs. Sin embargo, en [5] se hace uso de un criterio temporal para la obtención del modelo GMM que es necesario para posteriormente usarlo correctamente en la implementación conjunta con el RRT. Este hecho hace que no sea posible incorporar las distintas homotopías que pueden resolver la tarea en un mismo modelo. Nuestra aproximación presenta una mejora a este respecto, ya que de manera natural al integrar un planificador RRT*, la optimización de la función de costes en cada paso permite la selección automática de la homotopía más apropiada.

Relacionado con este trabajo, en [8] se presenta el método Informed-RRT*, el cual se centra en la búsqueda de las configuraciones más plausibles mediante el muestreo de un subconjunto de estados que minimicen la función de costes que se tiene en cuenta. Sin embargo este método utiliza una heurística encaminada a minimizar la longitud de las trayectorias propuestas, en lugar de minimizar el coste social (en general, molestias que puedan ocasionar los movimientos del robot en un entorno compartido con personas). De esta manera, en [8], tanto la función de costes como el subespacio de configuraciones no representan el comportamiento social exhibido por el robot durante la fase de aprendizaje, al contrario de nuestra aproximación.

El resto del documento se estructura del siguiente modo: la Sección 2 se ocupa de las tareas demostradas que se le enseñan al robot, mientras que la Sección 3 presenta los GMMs. La Sección 4 detalla cómo estos GMMs se pueden introducir en el planificador RRT*. Después, en la Sección 5 se detalla la configuración experimental, las métricas y simulaciones llevadas a cabo para validar tanto el aprendizaje de los modelos GMM como su integración con el planificador. Finalmente, en la Sección 6 se presentan las conclusiones y trabajo futuro.

2 Tareas Demostradas

Como se ha descrito anteriormente en la introducción, la capacidad de realizar una navegación social es crítica para el éxito en el despliegue de aplicaciones robóticas en las cuales se comparte espacio con las personas. A este respecto, en el

marco del Proyecto TERESA, vamos a analizar dos tareas en particular que permitan ilustrar cómo se comporta la aproximación propuesta. Por lo tanto, se le ha pedido a un usuario que realice las siguientes tareas un número determinado de veces mediante el pilotaje teleoperado del robot:

Evitación El robot esquiva a una persona parada y orientada hacia él. Esta maniobra de evitación puede ser realizada pasando tanto por la derecha como por la izquierda de la persona.

Aproximación El robot se aproxima a una persona parada que puede adoptar dos orientaciones relativas al robot: de frente o de espaldas a él. Cuando la persona mira directamente al robot, éste ejecuta el camino más corto para aproximarse a ella. Sin embargo, cuando la persona está de espaldas al robot, éste ejecuta caminos curvos de manera que evita acercarse por la espalda y procura entrar en el campo de visión de la persona antes de acercarse directamente a ella. En este caso también se contempla la realización de dicho acercamiento tanto por la derecha como por la izquierda de la persona.

La Fig. 1 muestra ambas tareas. Dichas tareas contemplan diferentes caminos o formas de alcanzar su objetivo: evitar o aproximarse a una persona. En este trabajo se consideran como homotopías los diferentes grupos de trayectorias que comparten un estado de configuración inicial y final en una determinada tarea pero que lo resuelven de manera diferente, comúnmente eligiendo caminos alternativos.

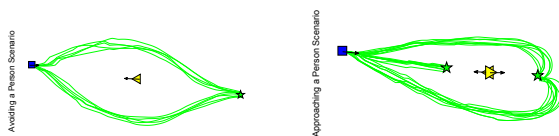


Figura 1: Homotopías ejecutadas.

3 Modelado HRI mediante GMM

Una elección apropiada de los indicadores $\mathbf{f} = [f_1, f_2, \dots, f_n]^T$ más relevantes cuando se codifica una tarea particular de navegación es crucial, más cuando esta selección provee parte de la solución del problema ya que define qué es importante a la hora de reproducir el comportamiento. En este trabajo se han considerado como indicadores la distancia (d) y el ángulo relativo (θ) entre el robot y la persona, en cada instante de

tiempo. El ángulo toma valores continuos en el intervalo $[-\pi, \pi)$, negativos en sentido de las agujas del reloj (respecto al frente de la persona). De este modo, consideraremos un conjunto compuesto por N puntos $\zeta = \{\zeta_j\}_{j=1}^N$ de dimensión $D = 2$, donde el indicador *tiempo* no se utilizará porque la dinámica de los comportamientos observados no se considerarán en este trabajo.

En un trabajo previo de los autores [14] se puede ver que los indicadores empleados ($d - \theta$) son suficientes para modelar las tareas en cuestión. Sin embargo, la selección de los indicadores no es una limitación de la técnica empleada, por lo que se deja para un trabajo futuro la inclusión de nuevos indicadores, incluyendo aquellos que pueden describir la dinámica de la ejecución de los ejemplos aprendidos, tales como el tiempo y las velocidades.

A partir de los indicadores extraídos de las trayectorias de ejemplo \mathcal{D} , es posible calcular un modelo GMM, de manera que la probabilidad de una combinación particular de estos queda dada por:

$$p(\mathbf{f}|\mathcal{D}) = \sum_{i=1}^k \omega_i \mathcal{N}(\mathbf{f}; \mu_{\mathcal{D}}^i, \Sigma_{\mathcal{D}}^i) \quad (1)$$

con k modos o Gaussianas. El modelo GMM queda definido por tanto por el conjunto de parámetros $\{\omega_i, \mu_{\mathcal{D}}^i, \Sigma_{\mathcal{D}}^i\}_{i=1}^K$, representando respectivamente las probabilidades *a priori*, los centros y covarianzas del modelo. Las probabilidades *a priori*, ω_i , satisfacen $\omega_i \in [0, 1]$ y $\sum_{i=1}^K \omega_i = 1$.

Los parámetros de la GMM son aprendidos mediante el algoritmo de *Expectación-Maximización* (EM) [6], el cual es inicializado con una estimación inicial de los centros y covarianzas realizados sobre los propios datos mediante la técnica de *k-means*.

Como desventaja del algoritmo EM, destacar que resulta necesario conocer de antemano el número de componentes k . Usualmente se sigue un criterio de selección basado en el Criterio de Información de Bayes (BIC) [15], el cual realiza un balance entre el ajuste del modelo a los datos y un factor de penalización que vela por una selección mínima de los modos del modelo. Para los experimentos presentados en este trabajo, hemos seleccionado empíricamente el mejor número de componentes que se ajustan a los datos de ejemplo.

4 El planificador para reproducción

RRT* [10] es una técnica para la planificación óptima (asintóticamente) de movimientos. Considera una función de costes asociada a cada punto x del espacio de configuraciones. El planificador

RRT* busca obtener la trayectoria ζ^* que minimice el coste total a lo largo del camino $c(\zeta)$. Esto lo hace mediante el muestreo aleatorio de dicho espacio de configuraciones y creando un árbol de estados desde el inicial hasta el estado final. Los caminos se representan pues por un conjunto discreto de configuraciones $\zeta = \{x_1, x_2, \dots, x_N\}$. Cada punto de dicha trayectoria puede así mismo asociarse a los valores de los indicadores calculados en cada punto, de manera que dicha trayectoria también puede describirse en el espacio de indicadores como $\zeta = \{f_1, f_2, \dots, f_N\}$.

Este trabajo extiende el algoritmo estándar RRT* con los modelos GMM aprendidos en tres niveles:

1. Incluir un nuevo coste de similitud en la evaluación de los nodos del árbol. El modelo GMM obtenido abarca las configuraciones más probables que se pueden dar en la tarea. Así, cuando un nodo se evalúa para ser añadido al árbol del RRT*, un coste basado en la GMM es usado para dicha evaluación. El objetivo es incrementar el coste de aquellas configuraciones menos probables, de acuerdo con el modelo. Para ello, la probabilidad se invierte para obtener el coste asociado. Es importante resaltar que dicha probabilidad es una densidad de ocurrencia, de manera que es necesario acotarla inferiormente para mantener su inversa en el intervalo $[0,1]$. Para ello, se ha elegido truncar la probabilidad a cierto valor inferior δ .
2. Proveer al planificador con el subespacio de estados más probable para realizar el muestreo del RRT*. Si el planificador conoce de antemano cuáles son los caminos más probables, entonces podemos sesgar el muestreo a dichas áreas, reduciendo la probabilidad de hacerlo sobre estados menos útiles y así, reducir los costes computacionales al obtener una solución. A este respecto, podemos especificar un porcentaje ajustable entre un muestreo uniforme y uno basado en el modelo GMM obtenido.

5 Montaje experimental y evaluación

5.1 Recopilación de datos para el aprendizaje

Se han llevado a cabo varios experimentos para recopilar las trayectorias de ejemplo que alimentarán la fase de aprendizaje de los modelos GMM. Dichos experimentos tuvieron lugar en una habitación despejada, libre de obstáculos entre las



Figura 2: Vista general de un momento del experimento.

personas allí presentes y el robot mientras se desarrollaron las tareas descritas en la sección 2. Usando un sistema de captura de movimientos (OptiTrack²) se extrajeron y se grabaron todos los datos referente a las posiciones y orientaciones del robot y las personas (ver Fig. 2).

5.2 Ensayos simulados

A partir de los datos recopilados, se extraen los modelos GMM de los indicadores utilizados utilizando el método descrito en la Sección 3, obteniendo un modelo para cada tarea. Posteriormente se integran dichos modelos en el planificador RRT* tal y como se explica en la sección 4. A este método conjunto lo llamamos *GMM – RRT**. Por otra parte, un conjunto auxiliar de modelos GMM en el espacio de configuración $x-y$ también se han extraído de los datos para ser utilizados en el planificador *GMM – RRT* [5], el cual fue originalmente propuesto para aprendizaje por demostración en planificación de movimientos para manipuladores, y que será empleado como base para comparaciones.

Para cada implementación de los planificadores se realizaron 25 simulaciones. Todas las planificaciones llevadas a cabo dispusieron de un tiempo de planificación de 100 segundos. Aunque este horizonte de simulación es demasiado elevado y no resulta adecuado para su utilización en aplicaciones realistas, se ha escogido para analizar la evolución de las métricas y costes durante las iteraciones de planificación, de modo que podamos comparar entre los distintos planificadores y parametrizaciones elegidas. También queremos analizar si existe algún comportamiento anómalo durante todo ese horizonte o tiempo de planificación establecido.

5.3 Métricas

Se proponen dos métricas para comparar los caminos obtenidos por los diferentes planificadores

²<https://www.optitrack.com/>

utilizados respecto de las trayectorias utilizadas como ejemplo:

La primera métrica se llama Métrica de Diferencia entre Trayectorias (Trajectory Difference Metric, TDM), que se define del siguiente modo:

$$TDM(\zeta_D, \zeta_P) = \frac{1}{|\zeta_D|} \sum_{i=1}^{|\zeta_D|} \min_{j=1}^{|\zeta_P|} \overline{\zeta_D(i)\zeta_P(j)} = \frac{1}{|\zeta_D|} \frac{1}{|\zeta_P|} \sum_{i=1}^{|\zeta_D|} \sum_{j=1}^{|\zeta_P|} \min_{k=1}^{|\zeta_P|} \overline{\zeta_D(i)\zeta_P(k)} \quad (2)$$

donde $\zeta_P(j)$ y $\zeta_D(i)$ son puntos de las trayectorias ζ_P y ζ_D a comparar, y $\overline{\zeta_D(i)\zeta_P(j)}$ es la distancia entre dos puntos. $|\zeta_D|$ y $|\zeta_P|$ son los números de muestras de cada trayectoria. Esta métrica nos da una idea de la similitud entre dos trayectorias dadas. Finalmente, se calcula la media entre todas las trayectorias de demostración y las planificadas:

$$TDM = \frac{1}{|D|} \frac{1}{|P|} \sum_{D,P} TDM(\zeta_D, \zeta_P) \quad (3)$$

donde los conjuntos D y P consisten en las trayectorias Demostradas y Planificadas, respectivamente.

El segundo indicador es la longitud de trayectoria resultante l_e , expresado como la media de la relación entre el valor absoluto de la diferencia entre las trayectorias planificadas y las demostradas dividido por la longitud de las trayectorias demostradas:

$$l_e = \frac{1}{|D|} \frac{1}{|P|} \sum_{D,P} \frac{|l(\zeta_D) - l(\zeta_P)|}{l(\zeta_D)} \quad (4)$$

5.4 Resultados

5.4.1 Evaluación de los costes

En los primeros experimentos se ha evaluado la velocidad de convergencia de los diferentes planificadores RRT^* en la consecución del camino óptimo, en términos de coste. La figura 3 muestra la evolución de dicho coste en función del número de iteraciones, para dos planificadores $GMM-RRT^*$: uno usando un 100% de muestreo GMM y el otro sin incorporar muestreo a este respecto (todo el muestreo es 100% uniforme, por tanto). El tiempo u horizonte de planificación para esta comparación fue de 100 segundos, que

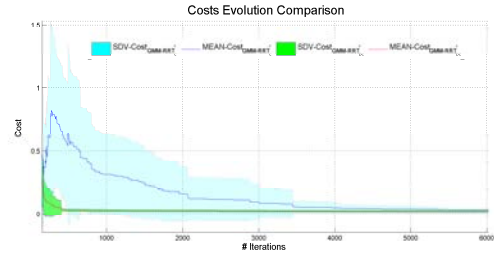


Figura 3: Comparación de la velocidad de convergencia para los planificadores $GMM-RRT^*$ con y sin muestreo basado en el modelo GMM obtenido. El gráfico muestra el valor medio y la desviación estándar obtenidos sobre todo el conjunto de simulaciones.

a pesar de ser excesivo es interesante de cara a tener un amplio margen de evaluación del comportamiento en cuanto a los costes. Puede verse en este ejemplo como el planificador que utiliza un muestreo basado en el modelo GMM obtenido de la tarea es hasta 100 veces más rápido que el que usa muestreo uniforme únicamente. Esto es obvio desde un punto de vista computacional ya que el espacio de configuración muestreado está muy cercano a la solución correcta, aprendida del modelo GMM, pero no esperábamos tal mejora.

5.4.2 Rendimiento de las métricas

Para los siguientes resultados se utilizó una estrategia de muestreo mixto: el 95% del tiempo se empleó muestreo basado en el modelo GMM aprendido, mientras que el restante 5% se empleó muestreo uniforme. Con esto intentamos sacar provecho de los modelos obtenidos y además dejar cierto grado de aleatoriedad en el sampleo de configuraciones, lo cual permite a nuestra aproximación tener un mejor rendimiento en algunas situaciones (ver 5.4.4). El tiempo de planificación empleado fue de nuevo 100 segundos. La figura 4 muestra un conjunto de 25 simulaciones para cada tarea, homotopía y planificador, con la parametrización explicada anteriormente.

Las tablas 1 y 2 muestran los valores obtenidos para las distintas métricas estudiadas, las tareas y los planificadores empleados.

Puede observarse que en ambas tablas el planificador propuesto en el presente trabajo es mejor para casi todas las simulaciones propuestas. Valores más pequeños (en negrita) significan mejores resultados.

Es importante resaltar que el rendimiento obtenido por el planificador $GMM-RRT$ es alcanzado en un tiempo medio de ejecución de 0.006 segundos, mientras que la optimización y

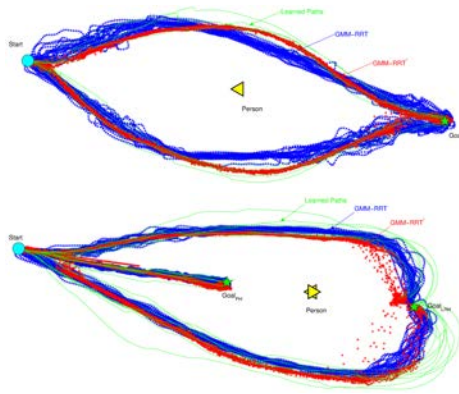


Figura 4: Las trayectorias demostradas y planificadas están dibujadas para cada tarea. Los estados “Start” y “Goal” también se indican, junto con la posición y orientación de la “Persona”, representada como un triángulo. Se ha empleado una estrategia de muestreo mixto para el planificador $GMM - RRT^*$ de 95%/5% (GMM y uniforme). El tiempo de planificación fue de 100 segundos. Arriba: Tarea de esquivar a una persona. Abajo: tarea de acercarse a una persona.

Tarea 1: Esquivar persona		
Planificador	TDM (m)	l_e (%)
Homotopía Derecha		
$GMM - RRT^*$	0.0565 ± 0.0152	3.66 ± 2.64
$GMM - RRT$	0.0803 ± 0.0205	4.00 ± 2.51
Homotopía Izquierda		
$GMM - RRT^*$	0.0484 ± 0.0109	4.96 ± 2.37
$GMM - RRT$	0.0676 ± 0.0169	4.31 ± 3.1

Tabla 1: Valores de las métricas para la tarea 1.

Tarea 2: Acercarse a persona		
Planificador	TDM (m)	l_e (%)
Homotopía Derecha		
$GMM - RRT^*$	0.0906 ± 0.0176	3.69 ± 2.54
$GMM - RRT$	0.0917 ± 0.0161	4.96 ± 3.45
Homotopía Izquierda		
$GMM - RRT^*$	0.0612 ± 0.0218	4.22 ± 2.50
$GMM - RRT$	0.0666 ± 0.0169	4.56 ± 3.62
Homotopía Frontal		
$GMM - RRT^*$	0.0376 ± 0.0141	11.35 ± 5.35
$GMM - RRT$	0.0337 ± 0.0184	8.55 ± 3.34

Tabla 2: Valores de las métricas para la tarea 2.

ejecución del planificador $GMM - RRT^*$ se extiende durante el tiempo de planificación empleado de 100 segundos. A continuación, la figura 5 muestra la evolución temporal de estas mismas métricas evaluadas en distintos tiempos de planificación (1.5, 3.0 y 5.0 segundos) para el $GMM - RRT^*$, en comparación con las obtenidas por el $GMM - RRT$ (primera columna).

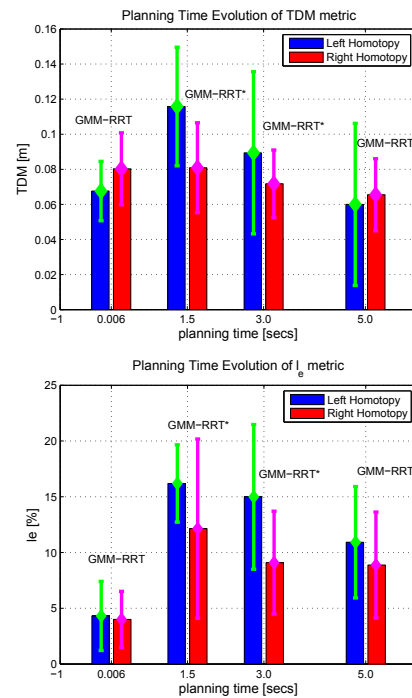


Figura 5: Media y desviación estándar de las métricas para distintos valores de tiempo de planificación. Tarea evaluada: acercarse a una persona.

Se ha elegido un tiempo de planificación de 1.5 segundos ya que se ha visto que era el tiempo mínimo que se invertía en esta tarea para encontrar una solución por parte del $GMM - RRT^*$. Como límite superior se ha optado por 5.0 segundos, que es un tiempo aceptable de planificación para la aplicación estudiada en este trabajo. En estos casos, el rendimiento obtenido por el $GMM - RRT$ mejora para todos los casos respecto a la métrica l_e (figura 5 -derecha), mientras que el $GMM - RRT^*$ sólo mejora la métrica TDM para un tiempo de planificación de 5.0 segundos.

5.4.3 Manejo de las homotopías

Como se comentó en la Sección 2, las tareas aprendidas por el robot están compuestas de varias homotopías. La figura 6 ilustra el modelo para la tarea de acercarse a una persona que se han empleado en el planificador $GMM - RRT^*$ propuesto. Este modelo, único, abarca tres homotopías. Para la misma tarea, si se pretende usar el otro planificador $GMM - RRT$, se hace necesario derivar un modelo por cada homotopía (figura 6).

Para entender esta necesidad de crear un modelo por cada homotopía se sugiere al lector que consulte el trabajo de referencia [5], concretamente la restricción temporal a la hora de encontrar el modelo GMM que define a la tarea.

Este último aspecto representa una clara desven-

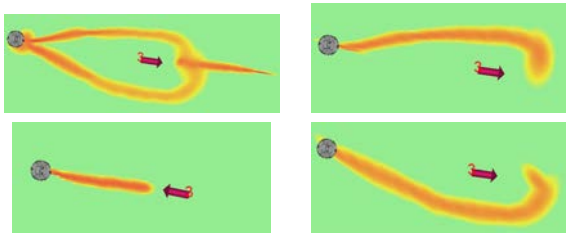


Figura 6: Modelos para la tarea de acercarse a una persona. Arriba-Izquierda: empleado en $GMM - RRT^*$, incluye las 3 homotopías. Resto: empleado en $GMM - RRT$, un modelo por cada homotopía.

taja del planificador $GMM - RRT$. No sólo ha de saberse de antemano la situación social en la que se encuentra el robot para elegir entre las homotopías aprendidas, sino que también puede ocurrir que un obstáculo llegue a impedir la ejecución de dicho plan. En esas situaciones, ambos planificadores pueden hacer uso de la variabilidad del modelo para conseguir un plan que llegue a esquivar dicho obstáculo, pero puede resultar que no sea posible conseguirlo. La figura 7 ilustra cómo los planificadores aquí comparados abordan este problema, en el caso concreto de estar realizando la tarea de evitar a una persona.

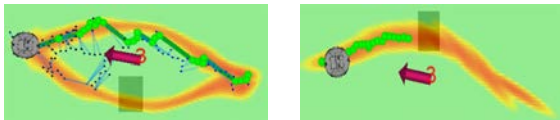


Figura 7: Las áreas sombreadas representan un obstáculo. Izquierda: el modelo incluye 2 homotopías y se usa el planificador $GMM - RRT^*$ durante un tiempo de 3 segundos. Derecha: Ejecución del plan obtenido por el $GMM - RRT$ teniendo en cuenta únicamente la homotopía derecha. Tiempo de simulación empleado de 100 segundos.

En el ejemplo expuesto en la figura 7 puede observarse cómo el planificador $GMM - RRT$ no es capaz de encontrar un camino libre hasta la meta, aunque exista. Sin embargo, el $GMM - RRT^*$ es capaz de elegir de manera natural entre las homotopías libres para alcanzar la meta. En caso de que no exista ninguna homotopía libre que conecte los estados inicial y final, el $GMM - RRT^*$ todavía es capaz de encontrar un camino gracias a la estrategia del muestreo mixto: el muestreo uniforme permite explorar de manera aleatoria el espacio de configuraciones que quedan fuera del modelo propio de la tarea, solventando de esta manera el citado problema.

5.4.4 Generalización

Con respecto al comentario anterior de la ventaja del uso de un muestreo mixto en el caso de la aproximación propuesta en este trabajo, el $GMM - RRT^*$, realizamos una primera evaluación que puede verse ilustrada en la figura 8: el escenario de acercarse a una persona es modificado y un obstáculo es introducido de manera que no exista un camino libre entre el inicio y la meta contenido en las homotopías. Esta situación no fue contemplada en la fase de aprendizaje y puede verse como una generalización del método propuesto.

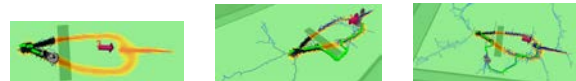


Figura 8: Izquierda: simulación sin muestreo uniforme. Centro: simulación con 5% muestreo uniforme. Derecha: simulación con 20% muestreo uniforme.

Todas las simulaciones de la figura 8 fueron realizadas con un horizonte de planificación de 100 segundos. Puede apreciarse como el planificador que utiliza un 100% de muestreo basado en el modelo GMM no es capaz de encontrar un camino libre hasta la meta, en dicho horizonte de tiempo empleado. Sin embargo, utilizando una estrategia de muestreo mixto, sí que se consigue progresar hacia la meta satisfactoriamente (figuras del centro y derecha).

En vista de estos resultados, los cambios adoptados en este trabajo sobre el planificador RRT^* (ver sección 4) todavía garantizan la integridad probabilística propia del algoritmo estándar. Aunque para el caso de que el muestreo sea 100% basado en el modelo GMM no haya alcanzado la meta en el caso presentado, el área verde del escenario mantiene una pequeña probabilidad de ser muestreada, de manera que en el caso de extender al límite el horizonte de planificación (infinito) el planificador acabará por encontrar una conexión entre ambos estados.

6 Conclusiones y trabajo futuro

En este trabajo presentamos un planificador basado en el conocido RRT^* . Mediante esta nueva aproximación, es capaz de planificar los caminos socialmente más aceptados en dos situaciones o tareas concretas. Cada tarea es caracterizada y modelada usando sendos GMMs, que posteriormente se utilizan de dos maneras: para guiar el muestreo del espacio de configuración y como coste a minimizar durante la obtención de los no-

dos o estados que conformarán el camino.

Conjuntamente se evalúa, bajo los mismos criterios, un algoritmo del estado del arte basado en *RRT* y *GMMs*. Este algoritmo resulta ser más rápido, y consigue alcanzar un mejor rendimiento de las métricas en menos tiempo. Sin embargo, las mejoras a este respecto son de 4 cm para la métrica TDM y de un 12% (60 cm) para la métrica l_e , en el caso peor.

No obstante, en cuanto a la flexibilidad de ambas propuestas y su capacidad de generalización, el planificador *RRT** supera claramente al anterior ya que 1) un mismo modelo puede incluir varias homotopías, 2) la selección de homotopías se hace de manera natural en el proceso de optimización que sigue el planificador y 3) la estrategia de permitir un cierto porcentaje de muestreo uniforme permite generalizar a nuevas situaciones.

Como trabajo futuro se plantea evaluar el uso de nuevos indicadores como el tiempo y la velocidad (cinemática del comportamiento aprendido). También aspiramos a crear una librería de comportamientos que se puedan usar combinadamente para acometer tareas más complejas.

Referencias

- [1] P. Abbeel and A. Y. Ng. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the twenty-first international conference on Machine learning, ICML '04*, pages 1–, New York, NY, USA, 2004. ACM.
- [2] B. Argali, S. Chernova, M. Veloso, and B. Browning. A survey of robot learning from demonstrations. *Robotics and Autonomous Systems*, 57:469–483, 2009.
- [3] S. Calinon. *Robot Programming by Demonstration: A Probabilistic Approach*. EPFL/CRC Press, 2009. EPFL Press ISBN 978-2-940222-31-5, CRC Press ISBN 978-1-4398-0867-2.
- [4] S. Calinon and A. Billard. A probabilistic programming by demonstration framework handling constraints in joint space and task space. Proc. IEEE/RSJ Intl Conf. on Intelligent Robots and Systems (IROS), 2008, 2008.
- [5] J. Claassens. An rrt-based path planner for use in trajectory imitation. In *ICRA*, pages 3090–3095. IEEE, 2010.
- [6] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *JOURNAL OF THE ROYAL STATISTICAL SOCIETY, SERIES B*, 39(1):1–38, 1977.
- [7] G. Ferrer, A. Garrell, and A. Sanfeliu. Robot companion: A social-force based approach with human awareness-navigation in crowded environments. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 1688–1694, Nov 2013.
- [8] J. D. Gammell, S. S. Srinivasa, and T. D. Barfoot. Informed rrt*: Optimal incremental path planning focused through an admissible ellipsoidal heuristic. *CoRR*, abs/1404.2334, 2014.
- [9] P. Henry, C. Vollmer, B. Ferris, and D. Fox. Learning to navigate through crowded environments. In *ICRA '10*, pages 981–986, 2010.
- [10] S. Karaman and E. Frazzoli. Sampling-based algorithms for optimal motion planning. *The International Journal of Robotics Research*, 30(7):846–894, 2011.
- [11] M. Luber, L. Spinello, J. Silva, and K. Arras. Socially-aware robot navigation: A learning approach. In *IROS*, pages 797–803. IEEE, 2012.
- [12] B. Okal, H. Gilbert, and K. O. Arras. Efficient inverse reinforcement learning using adaptive state-graphs. In *Learning from Demonstration: Inverse Optimal Control, Reinforcement Learning and Lifelong Learning Workshop at Robotics: Science and Systems (RSS)*, Rome, Italy, 2015.
- [13] N. Perez-Higueras, R. Ramon-Vigo, F. Caballero, and L. Merino. Robot local navigation with learned social cost functions. In *Informatics in Control, Automation and Robotics (ICINCO), 2014 11th International Conference on*, volume 02, pages 618–625, Sept 2014.
- [14] R. Ramon-Vigo, N. Perez-Higueras, F. Caballero, and L. Merino. *Analyzing the Relevance of Features for a Social Navigation Task*, pages 235–246. Springer International Publishing, Cham, 2016.
- [15] G. Schwarz. Estimating the dimension of a model. *The Annals of Statistics*, 6:461–464, 1978.
- [16] P. Trautman and A. Krause. Unfreezing the robot: Navigation in dense, interacting crowds. In *IROS*, pages 797–803. IEEE, 2010.