






Jornadas de Automática

Aprendizaje por demostración mediante datos sintéticos para tareas parametrizadas

Prados, A. , Mora, A. , Mendez, A. , Garrido, S. , Barber, R. 

RoboticsLab, Departamento de Ingeniería de Sistemas y Automática, Universidad Carlos III de Madrid, Av. Universidad, nº30, 28911, Leganés, España.

To cite this article: Prados, A., Mora, A., Mendez, A., Garrido, S., Barber, R. 2024. Learning from demonstration through synthetic data for parameterized tasks. *Jornadas de Automática*, 45.
<https://doi.org/10.17979/ja-cea.2024.45.10781>

Resumen

El aprendizaje de tareas por demostración es una tarea desafiante. Uno de los principales desafíos es la extrapolación del conocimiento a partir de demostraciones del usuario. Generar un conjunto de situaciones diversas cubriendo todos los casos posibles es una tarea compleja. Por esto, surge la idea de utilizar la parametrización de tareas, que genera información relevante de puntos característicos de una tarea, permitiendo modelar una política con una menor cantidad de demostraciones. En este artículo, presentamos un algoritmo centrado en generar información sintética para la generalización de tareas parametrizadas. El algoritmo permite la generación autónoma de datos, produciendo demostraciones de características similares a las de los usuarios. Para ello, una métrica basada en la distancia de Wasserstein, que tiene en cuenta los datos probabilísticos de las trayectorias aprendidas, ha sido desarrollada. Se han realizado pruebas en simulación, comparando la eficiencia con dos algoritmos de Lfd y en un entorno real, para una tarea de barrido, realizada con el robot ADAM.

Palabras clave: Robot móvil, Robot manipulador, Co-aprendizaje/Auto-aprendizaje, Aprendizaje por demostración, Imitación

Learning from demonstration through synthetic data for parameterized tasks

Abstract

Learning tasks by demonstration is a challenging task. One of the main challenges is extrapolating knowledge from user demonstrations. Generating a diverse set of situations covering all possible cases is a complex task. Hence, the idea of using task parameterization arises, which generates relevant information about characteristic points of a task, allowing the modeling of a policy with fewer demonstrations. In this paper, we present an algorithm focused on generating synthetic information for the generalization of parameterized tasks. The algorithm enables the autonomous generation of data, producing demonstrations with characteristics similar to those of users. For this purpose, a metric based on Wasserstein distance, which takes into account the probabilistic data of the learned trajectories, has been developed. Tests have been conducted in simulation, comparing efficiency with two LfD algorithms, and in a real environment, for a sweeping task, carried out with the ADAM robot.

Keywords: Mobile robots, Robots manipulators, Co-Learning/Self-learning, Learning from Demonstration, Imitation Learning

1. Introducción

Hoy en día, se espera que los robots tengan la capacidad de tener una alta autonomía, de operar en entornos dinámicos y de ser adaptables a diversas tareas. El Aprendizaje por Demostración (LfD) es una técnica prometedora, permitiendo a los robots adquirir diferentes habilidades motoras sin necesidad de una programación explícita. El LfD se basa en la

capacidad del robot para aprender tareas implícitas, sus restricciones y requisitos (Ravichandar et al., 2020), permitiendo la transferencia de habilidades por parte de no expertos, extrayendo las características y patrones más relevantes de las demostraciones, dándole un capacidad de generalización para diferentes tareas.

Una de las técnicas más relevantes de LfD son los algoritmos que codifican las habilidades mediante la representa-

ción basada en trayectorias, denominados como *aprendizaje basados en trayectorias*, que se basan en replicar la política extraída del estudio de los patrones de movimiento en situaciones novedosas (Calinon and Lee, 2017). Un método muy usado son los *Modelos de Mezcla Gaussiana para Tareas Parametrizadas* (TP-GMM) (Calinon, 2016). Esta técnica permite la generalización del modelo aprendido al codificar los estados relevantes para la tarea y utilizarlos para la generación de movimientos en nuevas situaciones. Los parámetros de tarea (TP) son frames de referencia que describen los arreglos espaciales del escenario. Usando puntos de vista desde muchos frames de referencia, se crea una política que se adapta de mejor forma dependiendo de la situación. Para funcionar eficientemente, este método requiere recopilar múltiples y diferentes situaciones para entender las diferentes configuraciones espaciales de la tarea a resolver con el fin de generar un modelo significativo.

Para solventar estas limitaciones, presentamos un algoritmo basado en la generalización de tareas parametrizadas mediante la datos sintéticos. En lugar de solo hacer uso de demostraciones humanas, el algoritmo es capaz de generar una serie de nuevas demostraciones que tienen las mismas características y habilidades intrínsecas. Estas demostraciones sintéticas se agregan al conjunto de datos del usuario para mejorar el rendimiento del TP-GMM. Esto reduce la dependencias de datos humanos al poder genera cualquier número de demostraciones requeridas con características similares a las humanas y elimina las ambigüedades en la resolución de una tarea. Las principales contribuciones de este algoritmo son:

- Generación de datos sintéticos: el método genera sus propias demostraciones sin intervención humana, basadas en pocos datos iniciales generados por un usuario.
- Evaluación de los datos generados para verificar si son similares a los proporcionados por el demostrador. Para esto se estima la energía de las trayectorias generadas (y no solo las diferencias geométricas como implementan otros métodos), permitiendo la mayor generalización del algoritmo.
- Los datos generados sintéticamente son completos (siempre generan una solución) y carecen de mínimos locales. Además eliminan las ambigüedades que puedan existir en un movimiento (dos formas distintas de resolver la misma tarea).

Para demostrar su eficiencia, se ha realizado una comparación con dos métodos de LfD a partir de pocas demostraciones además de una prueba con un robot real para resolver una tarea de barrido, consiguiendo aprende la política óptima, usando solo 3 demostraciones y generando el resto sintéticamente.

2. Estado del arte

En los últimos años, ha surgido una extensa investigación en el aprendizaje y generalización de movimientos. Dentro de este campo, se han desarrollado diferentes aproximaciones probabilísticas para el uso del LfD Ravichandar et al. (2020) que ofrecen la ventaja de no solo estimar la trayectoria subyacente de las demostraciones, sino también codificar su variabilidad a través de la covarianza Silvério et al. (2018). Estos

métodos poseen la capacidad de transferir la política de los movimientos aprendidos a nuevos entornos o tareas aprovechando las distribuciones probabilísticas además de ser capaces de modelar la variabilidad en una trayectoria, siendo un factor relevante para la generalización del conocimiento.

Dentro de este campo, el Modelo de Mezcla Gaussian (GMM) es uno de los métodos más usados, ya que permite la estimación integral de trayectorias, capturando la variabilidad mediante la matriz de covarianza, el manejo de la alta dimensionalidad, factor en el que otros métodos sufren limitaciones o la flexibilidad a la adaptación de trayectorias, mediante las distribuciones gaussianas, permitiendo su adaptación a un nuevo entorno. A pesar de esto, presenta varios desafíos. El primero surge de la cantidad de datos para lograr un modelo adaptable a nuevas situaciones donde es crucial emplear un número significativo de demostraciones con variabilidad entre ellas. Además, si la distribución no es adecuada o si hay desequilibrios en la cantidad de datos en diferentes regiones del espacio, el rendimiento puede verse afectado, dificultando el proceso de generalización. Para abordar estos problemas, se han creado modelos de tareas parametrizadas Li et al. (2023).

Los parámetros de una tarea encapsulan variaciones y características únicas exclusivas de una tarea particular, permitiendo un control y adaptación específico en el aprendizaje. Dicha idea se aplica en los Modelos de Mezcla Gaussiana para Tareas Parametrizadas (TP-GMM) Calinon (2016), que extienden el uso del GMM tradicional al considerar parámetros específicos para una tarea. Los TP-GMM permiten la codificación de información relacionada con la tarea en los componentes de mezcla del GMM. Los parámetros de tarea pueden capturar diversos aspectos, como estilos de movimiento o orientaciones específicas, permitiendo que los TP-GMM pueden aprender y representar variaciones específicas de la tarea. Los TP-GMM siguen necesitando una gran cantidad de datos y condiciones diferentes que se pueden utilizar para generalizar de manera correcta. En los últimos años, se han realizado múltiples mejoras en el TP-GMM original. En Pervez and Lee (2018) se presenta un método donde se utiliza primitivas del movimiento con TP-GMM para resolver el problema de divergencia en la generalización que surge de la existencia de datos dispersos. En Hoyos et al. (2016) se propone un método que incrementa la habilidad de aprendizaje, remodelando el conjunto de datos con la generación de demostraciones con nuevos parámetros generados por 3 métricas diferentes. En otros trabajos, como Sena et al. (2019), se realiza una parametrización a través de matrices de pesos.

El método desarrollado en este trabajo se enfoca en mejorar tanto la obtención de la política como en la reducción de errores en la generación de nuevas tareas. Para esto, el trabajo presentado usa un algoritmo de generación de datos sintéticos que optimiza automáticamente el modelo mediante la adición de los nuevos datos con características similares a las que podría producir un humano. Esto elimina problemas relacionados con la escasez de información o dificultades en la generalización cuando se tienen zonas con datos aglomerados.

3. Descripción del algoritmo propuesto

En este trabajo, hemos desarrollado un algoritmo (Fig. 1) para generar datos sintéticos que se asemejen a los datos hu-

manos y así aumentar la cantidad de información para la generalización del modelo. Mediante una evaluación en términos energéticos de los datos sintéticos, el algoritmo selecciona cuáles de los datos generados da un resultado al menos tan bueno como los datos humanos. Si es así, el algoritmo aumenta el conjunto de entrenamiento original para mejorar la política del modelo, lo que permite aprender tareas parametrizadas usando el conocimiento previo y el adquirido sintéticamente. El método (Alg. 1) comienza adquiriendo las demostraciones por parte del usuario (Lopez et al. (2023); Prados et al. (2023a)). Para ello, recopilamos ξ demostraciones, que cumplen $\xi \geq 2$. Todas las ξ recopiladas se agrupan en \mathcal{D}_{init} , que se utiliza para el primer entrenamiento del TP-GMM. El número de ξ necesarios para un buen modelo depende de la complejidad de la tarea. El usuario puede elegir la cantidad deseada de demos que necesita (M_{demos}) para una tarea en específico.

Algorithm 1 Aprendizaje de tarea mediante datos sintéticos

Input: N^o demostraciones, M_{demos} , Iteraciones máximas, L

Output: Modelo final TP-GMM, \mathcal{P}_{final}

- 1: Obtención de conjunto de datos inicial $\mathcal{D}_{init} \leftarrow \xi$.
 - 2: Modelo inicial TP-GMM \mathcal{P} usando solo \mathcal{D}_{init} .
 - 3: Coste energético inicial: $\mathcal{E}_{cost}(\mathcal{P}, \mathcal{D}_{init})$ ▶ Sección 3.2
 - 4: Inicialización: $Iteration = 0$, $n_d = \xi$ and $\mathcal{D} = \mathcal{D}_{init}$
 - 5: **while** $n_d \leq M$ or $Iteration < L$ **do**
 - 6: $\mathcal{D}_{new} \leftarrow S_{syntheticFMLData}(\mathcal{D})$
 - 7: Unificación de datos: $\mathcal{D}' \leftarrow \mathcal{D}_{new} + \mathcal{D}$
 - 8: Re-entrenamiento del modelo \mathcal{P}' usando \mathcal{D}'
 - 9: Nuevo coste usando el nuevo modelo: $\mathcal{E}'_{cost}(\mathcal{P}', \mathcal{D}')$
 - 10: **if** $\mathcal{E}'_{cost} \leq \mathcal{E}_{cost}$ **then**
 - 11: $\mathcal{D} \leftarrow \mathcal{D}'$, $\mathcal{P} \leftarrow \mathcal{P}'$, $\mathcal{E}_{cost} \leftarrow \mathcal{E}'_{cost}$, $n_d = n_d + 1$
 - 12: **else**
 - 13: Resta de conjuntos de datos: $\mathcal{D} \leftarrow \mathcal{D}' - \mathcal{D}_{new}$
 - 14: **end if**
 - 15: **end while**
 - 16: **return** $\mathcal{P}_{final} \leftarrow \mathcal{P}$
-

Con el modelo inicial \mathcal{P} , es necesario obtener una función de coste para medir la relación entre el modelo humano inicial y los datos sintéticos generados. Para ello, hemos implementado una función de coste basada en la Distancia de Wasserstein Xiao et al. (2019)(explicado en detalle en la Sección 3.2). Con este coste inicial (\mathcal{E}_{cost}), comienza el proceso de búsqueda de datos sintéticos aplicando un algoritmo propio de LfD, Kinesthetic Fast Marching Learning (KFML) Prados et al. (2023b). Los datos sintéticos se agregan al conjunto de datos inicial, obteniendo un nuevo conjunto \mathcal{D}' con el cual se reentrena el modelo \mathcal{P}' , generando una nueva política para los datos anteriores y los sintéticos agregados. Con \mathcal{P}' , el algoritmo calcula un nuevo valor de coste \mathcal{E}'_{cost} , que se compara con el valor de coste anterior. Si \mathcal{E}'_{cost} es menor o igual que el anterior, los datos agregados tienen una eficiencia igual o más óptima que la demostrada por el usuario, por lo que conjunto de datos y el modelo se actualizan. Si \mathcal{E}'_{cost} es mayor que el anterior, el conjunto de datos no es tan eficiente como las demostraciones humanas, y los nuevos datos sintéticos se descartan. Esto se ejecuta hasta que el número de datos agregados sea mayor que los valores de M establecidos por el usuario o la iteración del algoritmo sea mayor que el límite L .

3.1. Diseño del modelo mediante TP-GMM

Los TP-GMM se han propuesto en las últimas décadas como un enfoque para la programación de robots mediante demostraciones. Estos modelos codifican de manera probabilística la relevancia de los frames de referencia, los cuales pueden cambiar durante la tarea. En TP-GMM, los parámetros de la tarea (TP) se tratan como P sistemas de coordenadas, definidos en el tiempo t por $\{b_j, A_j\}_{j=1}^P$, donde b_j y A_j representan el origen del j -ésimo frame de referencia y un conjunto de vectores base $\{e_1, e_2, \dots\}$ formando la matriz de transformación $A = [e_1 e_2 \dots]$. De manera general los TP se especifican de antemano por el operador, de acuerdo con la tarea y basado en el conocimiento previo del operador. Esto permite que todos los parámetros elegidos sean relevantes para resolver una tarea específica.

Cada demostración, $m \in 1, \dots, M$, contiene una cantidad de T puntos de dimensión D $\{\xi\} \in \mathbb{R}^{D \times T}$, que se codifican en P frames de referencia diferentes, lo que resulta en un tensor de tercer orden: $\mathbb{R}^{D \times T \times P}$. Este conjunto de datos está compuesto por P muestras de trayectorias proyectadas en P frames de referencia candidatos, correspondientes a matrices compuestas por observaciones de D dimensiones en T pasos de tiempo. Los parámetros del modelo TP-GMM se definen como $\{\pi_i, \{\mu_i^{(j)}, \Sigma_i^{(j)}\}_{j=1}^P\}_{i=1}^K$, donde π_i son los coeficientes de mezcla, que representan la contribución de cada gaussiana, $\mu_i^{(j)}$ y $\Sigma_i^{(j)}$ son el centro y la matriz de covarianza del i -ésimo componente gaussiano en el frame j en un TP-GMM con K componentes. El aprendizaje de los parámetros implica maximizar el logaritmo de la verosimilitud bajo la restricción de que los datos en los frames son relevantes, para lo que se aplica Expectativa-Maximización (EM) que actualiza iterativamente los parámetros del modelo hasta converger.

$$\mathbf{E}\text{-step: } \gamma_{t,i} = \frac{\pi_i \prod_{j=1}^P \mathcal{N}(X_t^j | \mu_i^{(j)}, \Sigma_i^{(j)})}{\sum_{k=1}^K \pi_k \prod_{j=1}^P \mathcal{N}(X_t^j | \mu_k^{(j)}, \Sigma_k^{(j)})} \quad (1)$$

$$\mathbf{M}\text{-step: } \pi_i = \frac{\sum_{t=1}^T \gamma_{t,i}}{T}, \quad \mu_i^{(j)} = \frac{\sum_{t=1}^T \gamma_{t,i} X_t^{(j)}}{\sum_{t=1}^T \gamma_{t,i}} \quad (2)$$

$$\Sigma_i^{(j)} = \frac{\sum_{t=1}^T \gamma_{t,i} (X_t^{(j)} - \mu_i^{(j)})(X_t^{(j)} - \mu_i^{(j)})^T}{\sum_{t=1}^T \gamma_{t,i}} \quad (3)$$

Donde el superíndice t representa las iteraciones de EM. El modelo aprendido se puede utilizar para reproducir nuevas trayectorias para nuevas situaciones (nuevas posiciones y orientaciones de frames de referencia), generalizando así la política óptima aprendida. Para esto, el modelo recupera una GMM en cada t calculando el producto de gaussianas transformadas:

$$\mathcal{N}(\mu_{t,i}, \Sigma_{t,i}) \propto \prod_{j=1}^P \mathcal{N}(A_{t,j} \mu_i^{(j)} + b_{t,j}, A_{t,j} \Sigma_i^{(j)} A_{t,j}^T) \quad (4)$$

Tras esto, es posible aplicar un proceso de regresión gaussiana (GMR), permitiendo reproducir la trayectoria en función del modelo aprendido para los nuevos parámetros de la tarea, aprovechando la función de densidad de probabilidad conjunta descrita por TP-GMM. De esta manera, podemos controlar la generación de movimientos de un manipulador robótico mediante el aprendizaje de una política de movimiento óptima utilizando TP-GMM y un conjunto de datos proporcionados por el usuario, junto con sus parámetros relevantes.

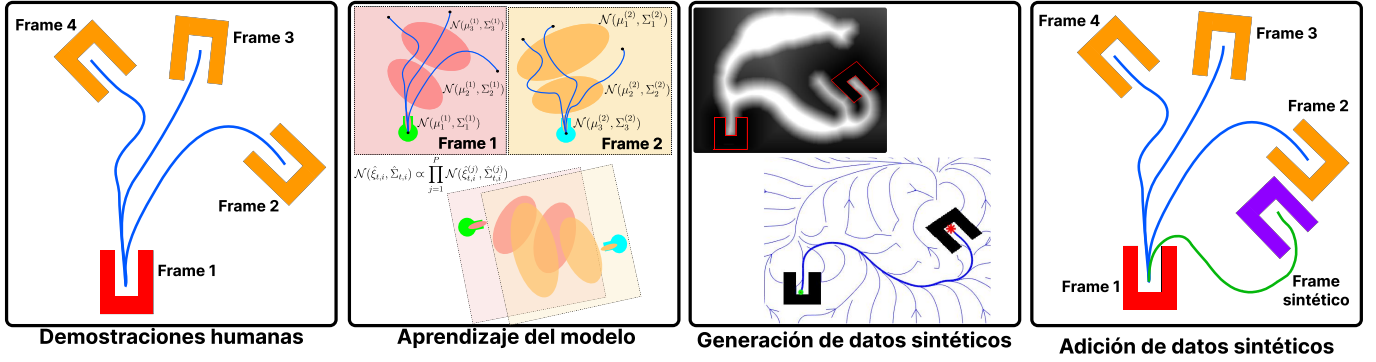


Figura 1: Esquema general del algoritmo. Se inicia con un conjunto pequeño de datos tomados por el usuario para generar un modelo inicial. Este modelo se utiliza como base para ejecutar datos sintéticos. Se evalúa su similitud, y en caso positivo, se agregan a los datos del modelo, mejorando así su generalización.

3.2. Función de coste basada en la distancia de Wasserstein

La distancia de p-Wasserstein es una métrica probabilística para cuantificar la disparidad entre dos distribuciones probabilísticas. Tiene en cuenta la manera en que las trayectorias de probabilidad deben ser "transportadas" para minimizar su coste. La distancia de p-Wasserstein se formula mediante los principios de la teoría del transporte óptimo. Al tener en cuenta la estructura geométrica de los espacios de probabilidad, la distancia de p-Wasserstein tiene la capacidad de capturar características distribuciones significativas, incluyendo la forma y la asimetría. Al trabajar en espacios continuos, se puede aplicar la norma L2 para la estimación de la distancia de p-Wasserstein, o 2-Wasserstein, definiendo nuestra función de coste, considerando la distancia cuadrada de 2-Wasserstein:

$$d(F_A, F_B) = \int_0^1 |F_A^{-1}(u) - F_B^{-1}(u)|^2 du = (\mu_A - \mu_B)^2 + (\sigma_A - \sigma_B)^2 + 2\sigma_A\sigma_B(1 - \rho^{A,B}) \quad (5)$$

donde la ubicación está representada por $(\mu_A - \mu_B)^2$ en la que μ_A y μ_B representan las medias de cada gaussiana, el tamaño está determinado por $(\sigma_A - \sigma_B)^2$ en la que σ_A y σ_B son las desviaciones estándar para cada gaussiana y la forma de cada gaussiana es generada por $2\sigma_A\sigma_B(1 - \rho^{A,B})$.

La función de coste \mathcal{E}_{cost} creada permite evaluar las similitudes entre distribuciones de probabilidad sobre el modelo TP-GMM usando la Ecuación 5 donde comparamos cada una de las gaussianas que determinan la trayectoria de la solución en términos de ubicación, tamaño y forma. Este coste considera toda la estructura de los espacios de probabilidad subyacentes de los TP-GMMs, ya que no solo evalúa la proximidad de los puntos, sino también la relación espacial y la disposición de las trayectorias. Adicionalmente, la sensibilidad a la forma de las distribuciones permite obtener sesgos importantes que no se tienen en cuenta solo con métricas espaciales. Es importante destacar su robustez frente a posibles valores atípicos en algunos de los datos y la invarianza a las transformaciones, lo que permite comparar trayectorias generadas que no sean exactamente iguales, permitiendo comparar diferentes datos generados sintéticamente.

3.3. Generación de datos sintéticos aplicando KFML

La generación de datos sintéticos (Alg. 2) comienza con la adición de datos del conjunto de demostraciones actual. El

algoritmo requiere la selección de datos por parte del usuario para el algoritmo KFML. Estos son el Área de Influencia (α) y la saturación (s), explicados en Prados et al. (2023b).

Algorithm 2 Generación de datos sintéticos mediante KFML

Input: Demostraciones: \mathcal{D}

Output: Solución generada sintéticamente: \mathcal{D}_{new}

- 1: Parámetros KFML: s, α \triangleright Elegidos por el usuario
- 2: Número demostraciones: $N_d \leftarrow \text{length}(\mathcal{D})$
- 3: $P_{final} \leftarrow \text{rand}(b_{min}, b_{max}), O_{final} \leftarrow \text{rand}(\beta_{min}, \beta_{max})$
- 4: Generación mapa: $map \leftarrow \text{binarize}(\mathcal{D})$
- 5: **for** $i = 0$ to N_d **do**
- 6: $demos\{i\} \leftarrow \text{interpolation}(\mathcal{D}, \text{size}(\mathcal{D}), \text{size}(map))$
- 7: **end for**
- 8: $F \leftarrow \text{KFML}(map, demos, s, \alpha)$ \triangleright Prados et al. (2023b)
- 9: $\mathcal{D}_{KFML} \leftarrow \text{pathGeneration}(F, P_{final}, O_{final})$
- 10: **return** $\mathcal{D}_{new} \leftarrow \mathcal{D}_{KFML}$

Tras esto, el algoritmo genera una posición final y una orientación dentro de un rango (restringido según los requisitos de la tarea). En nuestro caso, la orientación está limitada por $\beta \in [\beta_{min}, \beta_{max}]$ y la posición está limitada por $b \in [b_{min}, b_{max}]$. Una vez que se ha generado la posición del extremo, es necesario generar el mapa sobre el cual se realizará el proceso KFML (Fig. 2).

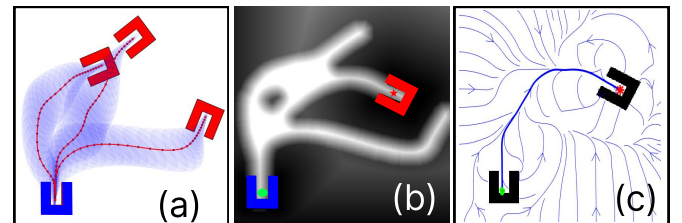


Figura 2: Ejemplo de generación de datos sintéticos. (a) Datos de demostraciones generados por humanos, (b) Mapa de velocidades para KFML usando una posición final aleatoria (punto rojo) con respecto a una posición inicial conocida (punto verde), (c) Datos sintéticos generados.

Este mapa no es más que una matriz sobre la cual aplicaremos el método. Esta matriz puede definirse por una imagen en la que se toman los datos (como puede ser en 2D) o por una matriz de mayor dimensionalidad. En nuestro caso, esta

matriz se genera a partir de los datos de demostración, por lo que se adaptará no solo al tamaño de los datos, sino también a la resolución de los datos, permitiendo su uso en casos 2D y 3D. El mapa generado se binariza donde 0 significa espacio libre y 1 significa obstáculo. Tras esto, se llama a KFML teniendo en cuenta el *map*, los *demos* y los valores de s y α . Esto permite obtener un mapa de velocidades F , sobre el cual podemos aplicar FM^2 como generador de trayectorias.

4. Resultados experimentales

4.1. Comparación de eficiencia con algoritmos de LfD

Nuestro algoritmo se ha comparado con otros dos algoritmos que buscan mejorar la generalización de los modelos con pocas demostraciones. El primer algoritmo, Zhu et al. (2022), se centra en la generación de un conjunto de datos sintéticos utilizando 3 técnicas para la generación de nuevos datos: agregar ruido blanco a los datos del experto (Noise), aplicar un proceso de generación de frames de referencia aleatorios que satisfagan las restricciones de la tarea para crear una nueva situación (RF) y la unión de ambos métodos (RF + Noise). El segundo, Sena et al. (2019), propone un método modificado de TP-GMM que considera la relevancia de los parámetros de la tarea durante la generación de trayectorias, determinada por la varianza de los datos. Para ese propósito, dicho algoritmo utiliza un método autónomo para ponderar los frames durante la regresión, mejorando la extrapolación del modelo, mientras se preserva la estructura local de los datos de demostración.

Para comparar los tres métodos (Fig. 3), se ha generado un entorno simulado $2m \times 2m$, con su centro en $(0, 0)$. El frame objetivo se posiciona y orienta aleatoriamente. El frame inicial permanece igual que en las demostraciones, ubicado en $(-0.8m, -0.8m)$ con una orientación aleatoria.

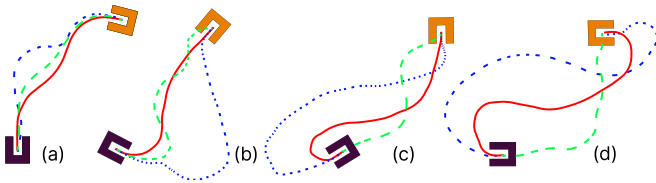


Figura 3: Resultados para 4 casos diferentes. Línea verde (dash grande), Zhu et al. (2022), línea azul (dash pequeño), Sena et al. (2019), línea roja (sólida), nuestro algoritmo, frame inicial (morada) y frame objetivo (naranja). (a) Frame inicial (0°), frame objetivo (85°), (b) Frame inicial (-110°), frame objetivo (120°), (c) Frame inicial (100°), frame objetivo (180°), (d) Frame inicial (90°), frame objetivo (-90°).

Los modelos generan una trayectoria que se evalúa por tres criterios: (i) longitud de la trayectoria, (ii) error en el punto final de la trayectoria y (iii) error de satisfacción de restricciones. La longitud de la trayectoria sirve como indicador de la calidad de la demostración, donde trayectorias muy largas pueden sugerir caminos incoherentes (comparándola con la media de los datos demostrados por el usuario). El error en el punto final de la trayectoria mide la distancia entre el punto inicial de la trayectoria y la ubicación del frame de inicio, así como la distancia entre el punto final de la trayectoria y la ubicación del frame objetivo. Esto evalúa la capacidad del modelo para generar trayectorias que comiencen y terminen en las

posiciones previstas. El error de satisfacción de restricciones evalúa la capacidad del modelo para producir trayectorias que se adhieran a la orientación correcta al entrar y salir de los frames de inicio y final (cajas U). Esta evaluación se realiza determinando el número de puntos de datos que caen dentro de las cajas U sin colisión, posicionadas en cada extremo de la trayectoria generada. El recuento de errores se calcula como el valor absoluto, asegurando que el método de aprendizaje sea penalizado tanto por una colisión con los límites de las cajas U como por una deficiencia de puntos de datos dentro de estas cajas U. Si se observan 10 puntos de datos dentro de cada caja delimitadora, se infiere que la trayectoria satisface con éxito la restricción de la tarea. Cada método se ha ejecutado 100 veces generando posiciones y orientaciones aleatorias para cada frame, utilizando el mismo conjunto de datos de demostraciones humanas y los resultados promedio han sido evaluados como una forma de comparar los diferentes métodos (Tabla 1). Todos los algoritmos funcionan correctamente cuando ambos frames están orientados en direcciones opuestas (Fig. 3a). Con orientaciones alteradas, el algoritmo presentado en Zhu et al. (2022) no puede reorientarse correctamente (Fig. 3c,d). El algoritmo presentado en Sena et al. (2019) es capaz de resolver todos los casos satisfactoriamente, independientemente de la orientación de los frames, pero se puede observar cuantitativamente que la ruta de solución es más larga que la generada como solución por nuestro algoritmo, por lo que, se generan resultados más óptimos en términos de longitud. Como se ve en la Tabla 1 cuantitativamente, el algoritmo genera soluciones más óptimas en término de longitud (genera paths más directos al punto final) y también tiene un error menor en el punto final que los otros dos métodos. Esto hace que las soluciones generadas por nuestro método sean más directas, suaves y eficientes (algo que se aprecia cualitativamente en la Fig. 3).

Tabla 1: Comparación para diferentes puntos finales e iniciales.

	Longitud		Error punto final		Error restricciones	
	Mean (m)	Std.	Mean (m)	Std.	Mean (m)	Std.
RF + Noise	3,98	$\pm 0,87$	0,080	$\pm 0,00$	18,75	$\pm 2,81$
α TP-GMR	5,73	$\pm 1,40$	0,040	$\pm 0,00$	1,00	$\pm 0,00$
Nuestro método	2,29	$\pm 0,53$	0,038	$\pm 0,00$	1,00	$\pm 0,00$

4.2. Experimentos entorno real: tarea de barrido

Para probar la eficiencia del algoritmo desarrollado en un entorno del mundo real, se ha considerado la tarea de barrido para su resolución utilizando nuestro método y la plataforma robótica ADAM Barber et al. (2022); Mora et al. (2024). Barrer es una tarea donde las limitaciones de orientación y posición son muy importantes, ya que los recogedores solo tiene un punto de acceso. Para el experimento, la suciedad es representada por bolas de papel que han de ser barridas al recogedor. La tarea será exitosa si la bola de papel entra en el recogedor con la orientación correcta y sin colisión (Fig. 4). Recopilamos 3 demostraciones (una para cada escenario) con una duración aproximada de 4 segundos. Los datos demostrados parten del mismo punto y acaban en diferentes puntos finales con diferentes orientaciones. Intencionalmente, solo se han demostrado tareas hacia una dirección de barrido (empujando) y siempre hacia la derecha del robot (orientaciones en las demostraciones entre 0° y 180°). Esto se ha hecho con el fin de probar las capacidades de generalización del algoritmo en ausencia de datos en el resto del espacio de trabajo.

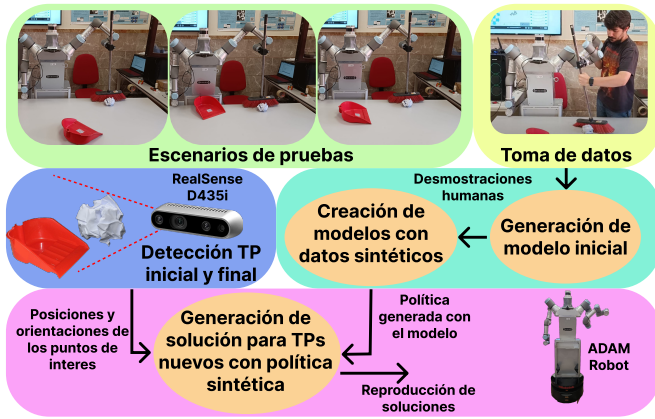


Figura 4: Esquema experimento real. El algoritmo únicamente hace uso de 3 demostraciones humanas y el resto son generadas sintéticamente.

El algoritmo hace uso de dos frames de referencia, uno ubicado en la bola de papel y el otro ubicado en el recogedor. Como datos de entrenamiento, se utilizan un número de demostraciones finales de $M = 10$, un número máximo de iteraciones de $L = 100$, y una cantidad de componentes gaussianos de $K = 9$. Para verificar la eficiencia del algoritmo, se han realizado diferentes variaciones tanto en la posición como en la orientación del punto inicial y final (Fig. 5). Los entornos de validación se han seleccionado para tratar de cubrir las diferentes formas de resolver la tarea de barrido, así como para probar la eficiencia contra datos diferentes a los de las demostraciones. Se observa que la bola de papel y el recogedor poseen posición y orientaciones diferentes a las de las demostraciones humanas. Los resultados mostrados en la Fig. 5 (inferior izquierda) son aquellos generados por el modelo entrenado con los datos de demostración más los generados sintéticamente. Nuestro algoritmo es capaz de generar trayectorias que resuelven satisfactoriamente las tareas de barrido independientemente de las posiciones y orientaciones del recogedor y la bola de papel. A pesar de tener únicamente datos humanos para recoger a la derecha del robot (Fig. 4, Escenarios de prueba), el algoritmo con los datos sintéticos, es capaz de generalizar y resolver tareas para las cuales a priori no tiene ninguna información. El vídeo disponible en YouTube presenta los experimentos además de tener el código en GitHub

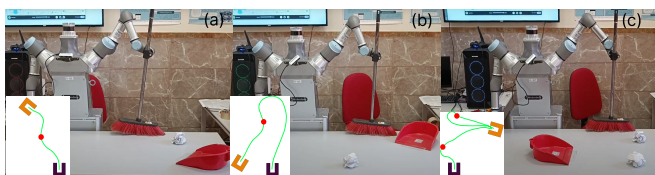


Figura 5: Entornos de validación para tareas de barrido con posición y orientación diferente para el papel y el recogedor. (a) Barrido mediante empuje. (b) Barrido mediante arrastre (c) Tarea con múltiples objetivos.

5. Conclusiones y trabajos futuros

El trabajo propuesto presenta un algoritmo de aprendizaje de tareas parametrizadas con un número reducido de demostraciones humanas. El algoritmo permite generar nuevas demostraciones que aumentan el conjunto de datos para mejorar el modelo. Para esto hace uso del KFML, un método

propio de LfD que permite generar datos solución con características humanas. Para la selección de resultados óptimos se ha usado la minimización de la energía basada en la distancia de Wasserstein, permitiendo la generación de trayectorias energéticamente similares a las demostradas. El algoritmo ha demostrado resolver eficientemente tareas bidimensionales y ha mejorado su eficiencia frente a otras técnicas similares de generalización, obteniendo resultados más eficientes. También se ha probado en una tarea de barrido en un entorno real con el robot ADAM basándose únicamente en 3 demostraciones humanas con una orientación similar. El algoritmo es capaz de generalizar a partir de datos sintéticos para lograr resultados satisfactorios.

Como trabajos futuros se propone detectar variaciones dinámicas en el entorno (como obstáculos dinámicos), considerándolos como parámetros a esquivar en la tarea durante la generación de la solución, permitiendo así poder realizar modificaciones rápidas en caso de ser necesario.

Agradecimientos

Este trabajo ha contado con el apoyo de Advanced Mobile dual-arm manipulator for Elderly People Attendance (AMME) (PID2022-139227OB-I00), financiado por el Ministerio de Ciencia e Innovación.

Referencias

- Barber, R., Ortiz, F. J., Garrido, S., Calatrava-Nicolás, F. M., Mora, A., Prados, A., Vera-Repullo, J. A., Roca-González, J., Méndez, I., Mozos, Ó. M., 2022. A multirobot system in an assisted home environment to support the elderly in their daily lives. *Sensors* 22 (20), 7983.
- Calinon, S., 2016. A tutorial on task-parameterized movement learning and retrieval. *Intelligent service robotics* 9, 1–29.
- Calinon, S., Lee, D., 2017. Learning control. In: *Humanoid robotics: A reference*. Springer Netherlands, pp. 1–52.
- Hoyos, J., Prieto, F., Alenyà, G., Torras, C., 2016. Incremental learning of skills in a task-parameterized gaussian mixture model. *Journal of Intelligent & Robotic Systems* 82, 81–99.
- Li, J., Cong, M., Liu, D., Du, Y., 2023. Enhanced task parameterized dynamic movement primitives by gmm to solve manipulation tasks. *RIA*.
- Lopez, B., Prados, A., Moreno, L., Barber, R., 2023. Taichi algorithm: Human-like arm data generation applied on non-anthropomorphic robotic manipulators for demonstration. In: 2023 ECMR. IEEE, pp. 1–7.
- Mora, A., Prados, A., Mendez, A., Espinoza, G., Gonzalez, P., Lopez, B., Muñoz, V., Moreno, L., Garrido, S., Barber, R., 2024. Adam: a robotic companion for enhanced quality of life in aging populations. *Frontiers in Neurobotics* 18, 1337608.
- Pervez, A., Lee, D., 2018. Learning task-parameterized dynamic movement primitives using mixture of gmms. *Int. Service Robotics* 11 (1), 61–78.
- Prados, A., López, B., Barber, R., Moreno, L., 2023a. Tracking humano visual aplicado a manipuladores no antropomórficos para imitación. In: XLIV Jornadas de Automática. Universidade da Coruña, pp. 714–719.
- Prados, A., Mora, A., López, B., Muñoz, J., Garrido, S., Barber, R., 2023b. Kinesthetic learning based on fast marching square method for manipulation. *Applied Sciences* 13 (4), 2028.
- Ravichandar, H., Polydoros, A. S., Chernova, S., Billard, A., 2020. Recent advances in robot learning from demonstration. *Annual review of control, robotics, and autonomous systems* 3, 297–330.
- Sena, A., Michael, B., Howard, M., 2019. Improving task-parameterised movement learning generalisation with frame-weighted trajectory generation. In: 2019 IROS. IEEE, pp. 4281–4287.
- Silvério, J., Huang, Y., Roza, L., Calinon, S., Caldwell, D. G., 2018. Probabilistic learning of torque controllers from kinematic and force constraints. In: 2018 IROS). IEEE, pp. 1–8.
- Xiao, H., Herman, M., Wagner, J., Ziesche, S., Etesami, J., Linh, T. H., 2019. Wasserstein adversarial imitation learning. *arXiv:1906.08113*.
- Zhu, J., Gienger, M., Kober, J., 2022. Learning task-parameterized skills from few demonstrations. *IEEE RAL* 7 (2), 4063–4070.