

Execution Fault Recovery in Robot Programming by Demonstration Using Multiple Models

J. Hoyos , F. Prieto, G. Alenya and C. Torras

Abstract— Deformable object (e.g., clothes) manipulation by a robot in interaction with a human being presents several interesting challenges. Due to texture and deformability, the object can get hooked in the human limbs. Moreover, the human can change their limbs position and curvature, which require changes in the paths to be followed by the robot. To help solve these problems, in this paper we propose a technique of learning by demonstration able to adapt to changes in position and curvature of the object (human limb) and recover from execution faults (hooks). The technique is tested using simulations, but with data obtained from a real robot.¹

Keywords— Robot programming by Demonstration, Robot Fault Recovery.

I. INTRODUCCIÓN

La programación por demostración (PpD) es una técnica que permite a un robot aprender la realización de una tarea, a través de la demostración de la misma por parte de un humano u otro robot [1], [2]. Se puede aplicar en entornos poco estructurados, porque permite que un usuario enseñe fácilmente nuevas tareas a un robot sin necesidad de conocimientos en programación. En PpD el robot es entrenado con ejemplos de la tarea, a partir de los cuales el aprende a generalizarla.

La tarea de poner una manga de camisa con un brazo robótico presenta una serie de retos interesantes. Por un lado, el robot debe operar cuidadosamente cerca de un humano, y la posición del brazo no siempre es la misma. Por otro lado, cuando se pone la manga de una camisa se producen frecuentemente enganches de la camisa en el codo. Una persona cuando pone la camisa detecta el enganche y de forma natural retrocede, modifica un poco la posición, y retoma la trayectoria cambiándola ligeramente. Este es precisamente el desempeño que se quiere obtener del robot.

Algunos de los retos en PpD en los que se investiga actualmente son: i) Técnicas que permitan generar nuevas trayectorias que guarden similitud con otras que fueron demostradas [3], [4], y ii) Técnicas que se recuperen ante fallos en la ejecución [5], por ejemplo agregando demostraciones correctivas a

un programa hecho manualmente [6]. En este trabajo se combinan estos dos tópicos.

Para el objetivo de la generación de nuevas trayectorias se propone utilizar la técnica de modelo de mezcla de gaussianas parametrizado en la tarea (en inglés “Task Parameterized Gaussian Mixture Model” que en este documento se abrevia como: TP-GMM) [4]. Cuando los robots manipulan objetos, sus movimientos pueden depender en gran medida de metas dadas y de las poses de los objetos, las cuales pueden ser definidas a través de marcos de referencia. Concretamente, el movimiento del robot es condicionado por un conjunto de variables de la tarea, que representan el sistema coordinado de marcos de referencia relevantes. A cambio de representar cada trayectoria como un modelo diferente, esta se basa en un solo modelo que abarca las diferentes trayectorias como función de las variables de la tarea, el modelo se basa en las propiedades del producto de gaussianas [7]. Para solucionar el problema planteado, la técnica TP-GMM es apropiada, por que genera la trayectoria requerida ante variaciones en la posición y curvatura del brazo, esta información del brazo es la que permite parametrizar la tarea.

La técnica propuesta consiste en un algoritmo que, a partir del conjunto de demostraciones con y sin fallo, aprende a recuperarse ante un fallo de ejecución (enganche). La técnica se compone de dos fases: i) estimación y ii) Ejecución. Además, permite agregar nuevos fallos.

Los resultados de la técnica se muestran a través de simulaciones, usando información de demostraciones reales obtenidas de manera kinestática con un robot WAM y un brazo de maniquí.

De la aplicación de vestir mediante robots, existen pocos trabajos reportados hasta la fecha. Algunos son [8] y [9]. En [8] se presenta una técnica en la que dos brazos robóticos, colocan entre sus propios brazos una camiseta. Los autores usan una técnica que combina aprendizaje por refuerzo, visión estéreo, y coordenadas topológicas, para obtener un modelo de la tarea compuesto por el estado del robot y el de la camiseta. Con un sistema de varias cámaras estéreo y marcas de color en las mangas obtienen la información que les permite calcular un modelo topológico de estas. A partir del modelo, la técnica de aprendizaje por refuerzo genera las trayectorias de las articulaciones que ponen la camisa, aun ante cambios de la forma de la camisa.

En [9], se presenta un trabajo similar al anterior, donde la tarea es colocar la camiseta sobre la cabeza de un maniquí. También usan aprendizaje por refuerzo y un modelo basado en coordenadas topológicas, tienen en cuenta para el modelo: i) la relación entre el cuello de la camisa y la cabeza del maniquí, ii) la relación entre el cuello de la camisa y el cuerpo del ma-

1J. Hoyos, Universidad del Quindío, Armenia, Colombia, josegabriell@uniquindio.edu.co F. Prieto, Universidad Nacional de Colombia, Bogotá, Colombia, faprieto@unal.edu.co

G. Alenya, Institut de Robotica i Informatica Industrial CSIC-UPC, Barcelona-Catalunya-España, galenya@iri.upc.edu

C. Torras, Institut de Robotica i Informatica Industrial CSIC-UPC, Barcelona-Catalunya-España, ctorras@iri.upc.edu

niquí y iii) la relación entre la manga y el brazo derecho del maniquí. Con esta información y mediante aprendizaje por refuerzo el robot logra colocar la camiseta.

Nuestro trabajo se diferencia de [8] y [9], en que éstos usan aprendizaje por refuerzo y además requieren un modelo en coordenadas topológicas de las mangas y hueco de la camiseta. También, con respecto a [9], el maniquí es fijo, a diferencia de nuestra propuesta que admite variaciones en posición y curvatura del brazo del maniquí. Por último, no toman en cuenta las fuerzas que se presentan en la tarea, para chequear el resultado de la colocación de la camiseta.

Lo que resta del documento presenta la siguiente estructura: en la Sección II se presenta la técnica propuesta, en la Sección III, se presenta el experimento realizado, posteriormente, en la Sección IV algunos resultados y su discusión. Finalmente, en la Sección V se presentan las conclusiones.

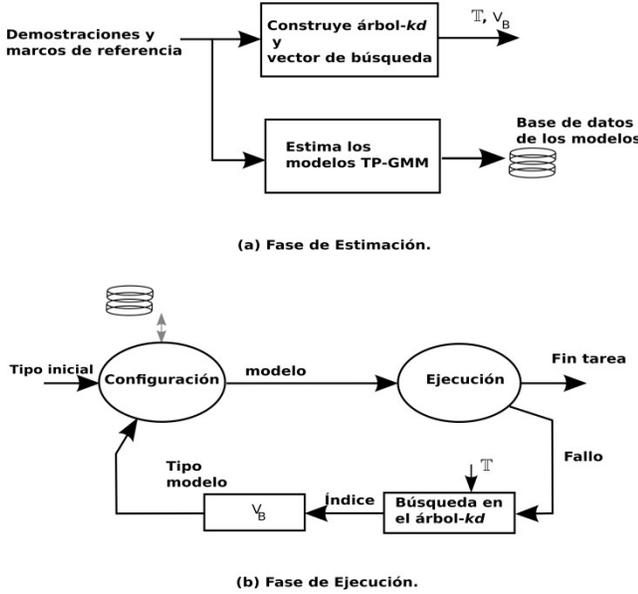


Figura 1: Diagramas de las fases de la técnica propuesta.

II. TÉCNICA PROPUESTA

En la Fig. 1, se muestran las dos fases del sistema propuesto. En la fase de estimación, con la información de todas las demostraciones, se construye un árbol-kd [10] y se estiman diversos modelos TP-GMM, que se obtienen al separar las demostraciones según la tarea que cumplen (colocar la manga de camisa, recuperar un fallo). En la fase de ejecución, dado un modelo TP-GMM, se generan los puntos de la trayectoria para la tarea normal (colocar la manga), al presentarse un fallo el árbol-kd es consultado y, con esta información, se obtiene el modelo TP-GMM que permite recuperarse del fallo de ejecución.

A. Fase de Estimación

La estimación se realiza a partir de dos conjuntos de demostraciones: i) las que elaboran la tarea sin fallos de ejecución y ii) las de recuperación, que son trayectorias que permiten recuperar el fallo y terminar la tarea.

Un conjunto de demostraciones compuesto de M trayectorias, cada una a su vez tiene T_m puntos de datos. Todos los puntos, forman una agrupación de datos $\{\xi_n\}_{n=1}^N$ con

$N = \sum_{m=1}^M T_m$, en el que cada punto se compone: $\xi_n = [t_n, y_n] \in R^{D+1}$, con D la dimensión de cada punto y t_n el instante de tiempo. Cada demostración se asocia con los parámetros de la tarea $\{A_{n,j}, b_{n,j}\}_{j=1}^{N_p}$ representando N_p marcos de referencia, con matrices de transformación $A_{n,j}$ y vectores de desplazamiento $b_{n,j}$. Los índices n y j representan la muestra de tiempo y el j -ésimo marco de referencia.

Además de lo anterior, para cada una de las trayectorias del conjunto se agregaron dos etiquetas: i) Conjunto de pertenencia y ii) Fallo Ocasional. La primera, contiene una etiqueta según el conjunto de pertenencia {Sin Fallo, Fallo 1}, la segunda contiene información de cuál fallo de ejecución ocurre ocasionalmente cuando se ejecuta. Con la primera se genera de manera automática un vector de tamaño $2 \times M$, al que denominamos vector de búsqueda, donde cada elemento contiene el modelo usado.

A.1) Estimación del Árbol-kd

Con la información de todas las demostraciones y sus respectivos parámetros de la tarea, se construye un árbol-kd, el cual es una estructura que permite almacenar conjuntos de puntos (nodos) en un espacio k -dimensional y es una generalización del árbol de búsqueda binaria.

Un nodo del árbol-kd, está formado por los N_p parámetros de la tarea (sin el dato de tiempo). Todos los nodos forman la matriz de datos η , con la que se construye el árbol-kd: T , dando como resultado una indexación de los nodos por similitud en sus valores.

$$T = \text{build}_{kdtree}(\eta), \quad (1)$$

donde η , es de dimensión $(N_p \times D) \times (2 \times M)$.

A.2) Estimación del Modelo TP-GMM

Como las demostraciones están etiquetadas, se pueden separar nuevamente en conjuntos y estimar n_m modelos TP-GMM, con lo que se genera la base de datos de modelos. El vector de búsqueda V_B , generado a partir de las etiquetas de conjunto de pertenencia, permite convertir de índice de un nodo a modelo.

En la estimación de un modelo TP-GMM, las entradas son un conjunto pequeño de demostraciones y los parámetros de la tarea. Estos últimos son marcos de referencia de puntos que guardan alguna relación con cada trayectoria demostrada. Por ejemplo, en la Fig. 2 los puntos de referencia son mostrados como círculos rojos al final de cada trayectoria demostrada.

Una vez estimado el modelo, se obtiene como salida la información compuesta por los coeficientes de mezcla, centros, y covarianzas de las gaussianas, que son los que denominamos parámetros del modelo. Los parámetros del modelo son $\{\pi_i, Z_{i,j}^\mu, Z_{i,j}^\Sigma\}$, que representan el coeficiente de mezcla, el vector de promedios y la matriz de covarianza para cada

marco j y la componente de mezcla i . Con estos parámetros del modelo, se calcula el centro resultante $\mu_{n,i}$ y la matriz de covarianza $\Sigma_{n,i}$ de cada componente i , como el producto de las gaussianas linealmente transformadas:

$$N(\mu_{n,i}, \Sigma_{n,i}) = \prod_{j=1}^{N_p} N(A_{n,j} Z_{i,j}^\mu + b_{n,j}, A_{n,j} Z_{i,j}^\Sigma) \quad (2)$$

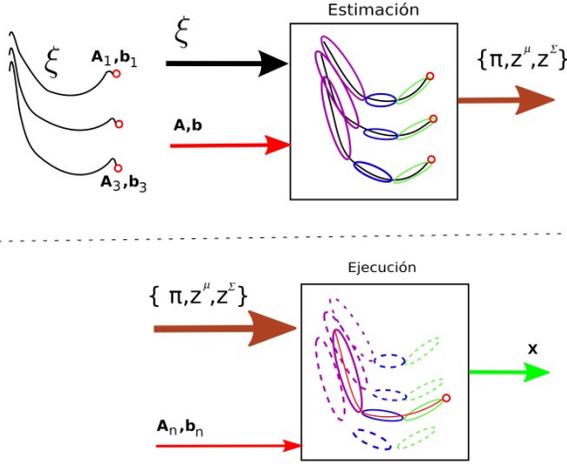


Figura 2: Diagrama de la técnica TPGMM. Parte superior: estimación del modelo. Inferior: cálculo de una nueva trayectoria.

Usando la propiedad del producto de distribuciones normales, el centroide y la matriz de covarianza de la ecuación anterior son:

$$\begin{aligned} \hat{\mu}_{n,i} &= \sum_{j=1}^{N_p} \hat{\mu}_{n,i,j}^{-1} (A_{n,j} Z_{i,j}^\mu + b_{n,j}), \\ \hat{\Sigma}_{n,i} &= \sum_{j=1}^{N_p} \hat{\Sigma}_{n,i,j} \end{aligned} \quad (3)$$

$$\begin{aligned} A_{n,j} Z_{i,j}^\Sigma A_{n,j}^T &= \hat{\Sigma}_{n,i,j}^{-1} \\ \hat{\Sigma}_{n,i} &= \hat{\Sigma}_{n,i,j} \end{aligned}$$

Los parámetros del modelo se estiman iterativamente usando una modificación del procedimiento Esperanza-Maximización (EM) [4],[7], [11]. Cada iteración consta de dos pasos, el primero calcula la probabilidad posterior del modelo y, el segundo, calcula los parámetros del modelo con base a esta probabilidad. La modificación con respecto al EM tradicional consiste en que el algoritmo tiene en cuenta los parámetros de la tarea [4]. La reproducción del modelo se realiza usando la técnica de regresión de mezcla de Gaussianas parametrizado en la tarea (TP-GMR del inglés “Task-Parametrized Gaussian Mixture Regression”) [4],[7].

A.3) Detección y Clasificación del Fallo

La detección se realiza a través de una condición programada manualmente. La condición consiste en que la suma de las fuerzas en los tres ejes del sensor fuerza/torque, no sobrepase un valor umbral. Por otro lado, con la etiqueta de fallo ocasional se genera, de forma automática, una tabla de fallos, con la

cual es posible saber el tipo de fallo o si se trata de un fallo desconocido.

B. Fase de Ejecución

El estado de configuración (Fig. 1b), es el que define que modelo usar para cierta tarea (colocar manga, recuperación). A partir de un dato de parámetros de la tarea P_t , se consulta al árbol-kd, para conocer cual es el nodo más cercano, obteniendo un índice i_x :

$$i_x = \text{knn}_{\text{near}}(P_t, T) \quad (4)$$

Este índice y el vector de búsqueda, permiten encontrar el tipo de modelo. Con este tipo, se extrae el modelo TP-GMM de la base de datos. Inicialmente, como se supone sin fallo, P_t se obtiene de los marcos de referencia para la tarea normal. En caso de fallo de ejecución, P_t también es obtenida con los marcos de referencia, pero según el tipo de fallo. El sistema pasa a estado de ejecución, donde se reproduce la trayectoria y se chequea la condición de fallo. Si existe un fallo, se vuelve al estado de configuración.

En ciertos casos, el dato P_t puede ser incorrecto, por lo que es necesario chequear la distancia Δ entre el nodo dado por el índice obtenido P_a y el dato suministrado:

$$\Delta = \|P_a(i_x) - P_t\| \quad (5)$$

si esta distancia es mayor a (0.12), se considera que no fue encontrado un modelo solución de la tarea y se cae en el caso de modelo desconocido. Este último caso también se presenta si después de un fallo, la solución dada es el modelo sin fallo.

C. Inclusión de la Recuperación de un Nuevo Fallo

La técnica permite la inclusión de la recuperación de un nuevo fallo, para esto se le agrega la información de un conjunto de nuevas trayectorias que recuperan del fallo, lo cual se realiza en la fase de estimación, con los siguientes pasos:

- El usuario agrega la información de las demostraciones y marcos de referencia, que permiten recuperar el nuevo fallo.
- El sistema incorpora la información en el árbol-kd y estima además un modelo TP-GMM, el cual se adiciona en la base de datos.
- El sistema vuelve a la fase de ejecución.

III. ADQUISICIÓN DE LA INFORMACIÓN

En la Fig. 3 se muestra una secuencia de imágenes de una demostración kinestática de la tarea de colocar la manga de la camisa. En este, una persona guía el efector final del robot con su mano, y se capturan los valores de articulación y pose de la trayectoria del robot colocando la manga. Para la adquisición de los datos se usó un robot WAM de siete grados de libertad y un brazo de maniquí. El efector final del robot tiene una pinza (Fig. 4) y un sensor de fuerza/torque (ATI-40). Con el sensor se adquieren las fuerzas en las tres coordenadas cartesianas, y los torques de los tres ángulos de orientación del efector final.

El brazo del maniquí es de alambre y espuma, forrado en tela, y permite variar la curvatura del codo del brazo y la posición espacial. Mediante una articulación en el hombro se puede gi-

rar de manera manual, para cambiar la posición hacia adelante o hacia atrás, como puede verse en la Fig. 5.

Este trabajo se centra en los aspectos de aprendizaje de la tarea y la estrategia de recuperación ante un fallo, por lo que la tarea de percepción se ha simplificado y los parámetros de la tarea se suministran manualmente al modelo, como se detalla en la siguiente sección.

Durante el experimento se usaron cuatro posiciones, las cuales fueron marcadas en la articulación del maniquí como H0.0, H1.0, H1.5 y H2.0 (Fig. 5), para hacer posible su posterior reproducción. El grado de curvatura del codo se varió entre curvatura tipo cero C0 (brazo extendido) y tipo tres C3 (aproximadamente 90 grados entre el antebrazo y el brazo).

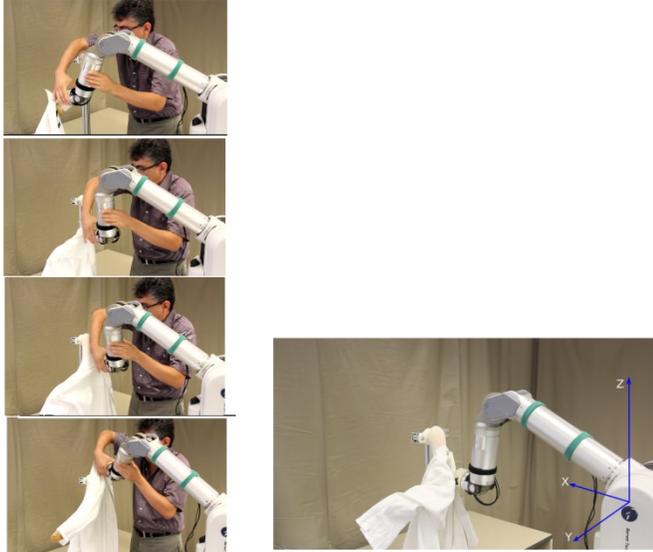


Figura 3: Demostración kinestática de una trayectoria. En color azul, el marco de referencia de la base del robot.

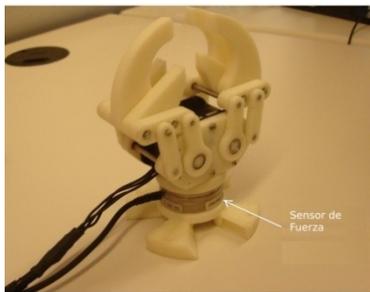


Figura 4: Imagen de la pinza y el sensor de fuerza/torque.

Para cada demostración de la tarea de colocar la manga de la camisa, se configuró manualmente la posición y curvatura del brazo del maniquí. El agarre de la camisa por parte del robot se realizó manualmente, aunque puede ser automatizada [12].

Además de las trayectorias, se tomaron puntos (mano, codo, hombro) en el brazo del maniquí, los cuales se usan para calcular los parámetros de la tarea. Toda la información y las trayectorias, esta referenciada a la base del robot.

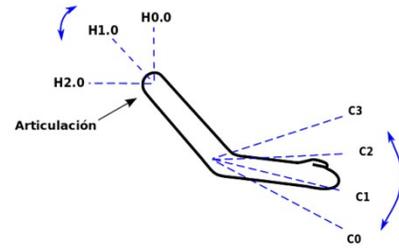


Figura 5: Diversas posiciones y curvaturas del brazo del maniquí.

TABLA I: COMBINACIONES DE POSICIÓN DE LA MANO Y CURVATURA DEL CODO QUE PRODUCEN LAS TRAYECTORIAS DEMOSTRADAS SIN FALLO.

Combinación	Descripción
H0.0 / C0	Brazo extendido con la mano en posición baja
H1.0 / C1	Brazo con curvatura de 120 grados con la mano en posición media
H0.0 / C3	Brazo cerrado a 90 grados con la mano en posición baja
H1.0 / C3	Brazo cerrado 90 grados con la mano en posición media
H1.5 / C3	Brazo cerrado 90 grados con la mano en posición media-alta
H2.0 / C3	Brazo cerrado 90 grados con la mano en posición más alta

A. Trayectorias sin Fallo

Al variar la posición de la mano y curvatura del brazo (Fig. 5), se puede inferir que se pueden producir diversas combinaciones. De estas, se seleccionaron de forma heurística seis, buscando que fueran muestras representativas, las cuales se listan en la Tabla I. Se aplicó una codificación con modelo de mezcla de gaussianas a tres demostraciones por cada ítem de la Tabla I. Las trayectorias resultantes de la codificación se muestran en la Fig. 6.

B. Trayectorias de Recuperación

En la Fig. 7 se muestran las trayectorias de recuperación de fallo de ejecución en el plano yz y xy, codificadas con la técnica GMM a partir de dos demostraciones de cada una de las combinaciones de posiciones y curvatura tipo tres. En esta gráfica también se pueden ver los parámetros de codo y enganche, mostrados con círculos de color violeta y verde, respectivamente.

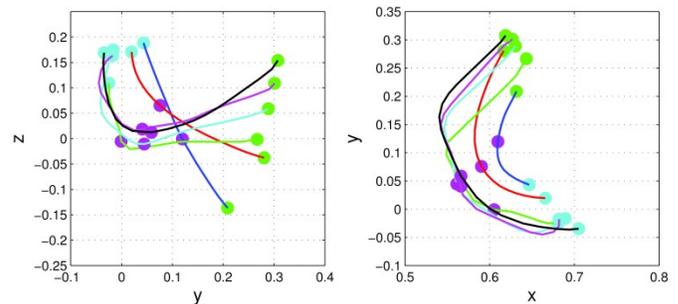


Figura 6: Trayectorias demostradas sin fallo, vistas desde los planos yz y xy. Los círculos en color (verde, violeta y azul), son los valores b en los parámetros de la tarea, para cada trayectoria.

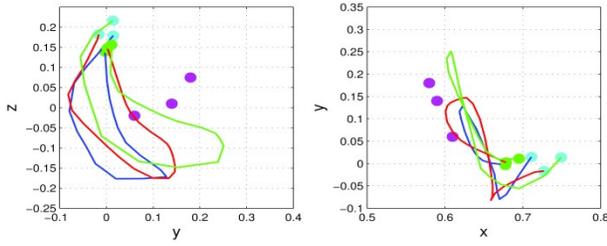


Figura 7: Trayectorias de recuperación de fallo de ejecución vistas desde los planos yz y xy. Los círculos en color (verde y violeta), son los valores de b en los parámetros de la tarea.

IV. SIMULACIÓN Y RESULTADOS

En esta sección se presenta la estimación de los modelos, se explica cómo se realiza la detección del fallo de ejecución, y se muestran resultados de la técnica propuesta, usando simulaciones en Matlab.

A. Estimación de los modelos

A.1) Cálculo de los Parámetros de la Tarea

Los parámetros de la tarea guardan la información sobre variaciones de posición y dirección de marcos de referencia, por ejemplo la mano o el hombro.

Los parámetros de la tarea están compuestos por la matriz de transformación A (dirección) y el vector de desplazamiento b (posición). Se deben calcular u obtener tanto en el proceso de estimación del modelo, como cuando se pretende calcular una nueva trayectoria. El cálculo de A y b se realiza a partir de los puntos p_1 y p_2 en el espacio cartesiano, referenciados a la base del robot WAM. Similar a [4] la matriz de transformación A , se forma a partir de tres vectores v_1 , v_2 y v_3 :

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & v_{1x} & v_{2y} & v_{3y} \\ 0 & v_{1y} & v_{2y} & v_{3y} \\ 0 & v_{1z} & v_{2y} & v_{3y} \end{bmatrix}. \quad (6)$$

El vector v_2 es el vector normal a la diferencia $p_2 - p_1$, y los vectores v_1 y v_3 son perpendiculares a v_2 , con lo que A es una matriz de transformación que contiene información de la orientación del punto p_1 con respecto al punto p_2 . El vector de desplazamiento b se forma a partir del punto p_1 , como:

$$b = \begin{bmatrix} 0 & p_{1x} & p_{1y} & p_{1z} \end{bmatrix}^T. \quad (7)$$

A.2) Detalles del Modelo Trayectorias Sin Fallo

En el primer modelo TP-GMM, el número de gaussianas se seleccionó de manera heurística en seis y se usaron tres parámetros de la tarea: el primero es la posición inicial de la trayectoria, la cual básicamente está sobre la mano; el segundo es la posición del codo; y el tercero la posición del hombro. Estos se seleccionaron porque el primero guarda la variación de la posición donde se encuentra la mano, el segundo permite conservar la relación que existe entre la trayectoria y el punto de

codo (curvatura), y el tercero suministra la posición del hombro.

Para el primer parámetro que corresponde a la mano, p_1 es un punto cartesiano que está a dos centímetros sobre la mano, y p_2 es el dato leído en el codo. En el caso del hombro, p_1 es la posición del hombro, y p_2 es la del codo. Para los parámetros anteriores, el cálculo de A y b se realizó como se explica anteriormente. Para el caso del codo, el vector de desplazamiento b se calculó con la posición del codo, y la matriz de transformación A se tomó como la matriz identidad de 4×4 . En todos los casos, los puntos p_1 y p_2 se obtuvieron de manera manual, guiando el efector final a la posición deseada y almacenándola.

A.3) Detalles del Modelo de las Trayectorias de Recuperación

En esta implementación se trabajó con curvaturas tipo tres, donde ocurren la mayoría de los enganches (Ver Fig. 5). En este caso, para estimar el modelo TP-GMM, solo se usaron tres trayectorias correspondientes a cambios de posición del codo.

Las trayectorias originales que reproducen la ruta de recuperación (regreso y avance), con las cuales se estima el

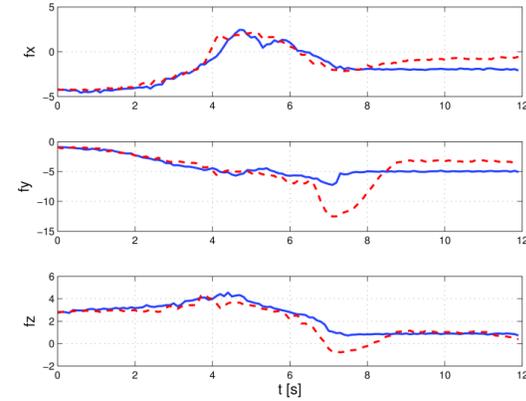


Figura 8: Dos casos de señales de fuerzas para la tarea de colocar la camisa. Azul/continua: La manga de la camisa logra ser colocada correctamente. En Rojo/punteada: Se presenta un enganche.

modelo, se obtuvieron a partir de demostraciones kinestáticas y la captura de la información de los parámetros se realizó manualmente, el punto de codo se tomó colocando el efector final del robot en el codo y grabando su valor. Los marcos de referencia de puntos de enganche de las tres demostraciones se eligieron similares a los observados en pruebas preliminares, donde ocurrieron enganches.

Se utilizaron seis estados o gaussianas y se seleccionaron tres parámetros de la tarea: el primero es un punto cercano al hombro (punto de enganche), el cual permite guardar la variación de punto de inicio de la nueva trayectoria; el segundo es el parámetro de codo, que permite que la trayectoria cambie al variar la posición del codo; y el tercero la posición del hombro. En todos, la matriz de transformación A se tomó por simplicidad como la identidad.

B. Detección del Fallo de Ejecución

En la Fig. 8 se presentan las fuerzas del sensor en el efector final del robot que permiten la detección del fallo. Las fuerzas en azul (línea continua), son las obtenidas durante una prueba

donde la manga logra ser colocada correctamente. En rojo (línea punteada), se presenta un enganche, y se puede observar que la fuerza en el eje y se incrementa más de 10 Newtons. Experimentalmente se encontró un valor del umbral de fuerza de 12 Newtons, cuando se presentan enganches.

C. Resultados

C.1) Simulación de Ejecución de Trayectorias (Casos Base)

Los casos base son aquellos en los que se emplea directamente los valores de los parámetros de la tarea, de las demostraciones usadas. En la Fig. 9, se muestra un ejemplo, donde se simulo que la fuerza no se aumentó en un tiempo dado, con lo que el sistema no detecta fallo. En rojo se muestra la trayectoria obtenida con TP-GMR y en negro la trayectoria original usada para estimar el modelo (curva C3 y posición H2.0). En la Fig. 10, se muestra la simulación donde la fuerza aumenta hasta el umbral de fallo, en un tiempo $t = 7.3s$, el cual es el tiempo promedio donde ocurren los enganches cerca al hombro. En esta figura, se puede ver como la técnica al ocurrir el enganche realiza la recuperación, la cual consiste en devolver la manga y regresar por una trayectoria diferente.

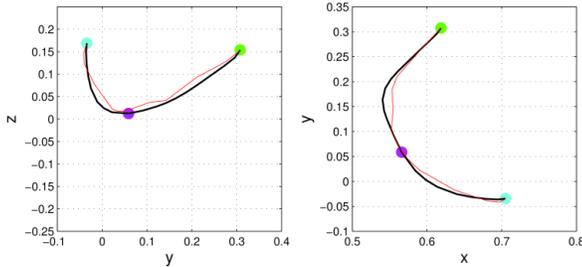


Figura 9: Ejemplo de simulación sin ocurrencia de fallo, trayectoria con curva C3 y posición H2.0.

Se calculó, además, la diferencia punto a punto [13] entre cada una de las seis trayectorias de entrenamiento (casos base sin fallo) y la simulación respectiva, dando un error RMS promedio de 1.2 % para los seis casos, y para los tres casos de recuperación del 2.2 %.

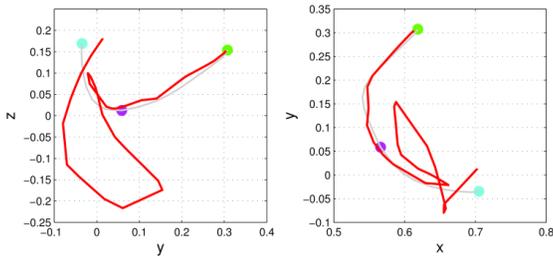


Figura 10: Ejemplo de simulación de la ocurrencia de fallo de ejecución.

C.2) Simulación Variando los Parámetros

Se evaluó el uso de valores de los parámetros de la tarea diferentes a los empleados en la estimación del modelo. En la Fig. 11, se muestra varios casos de simulación (a-d). En estos casos, el aumento de fuerza hasta el valor de umbral, se simulo en un tiempo $t = 7.3s$. En gris se muestran las trayectorias base más similares a estos nuevos casos en rojo la respuesta de la técnica propuesta. Los casos a-c no producen trayectoria de recuperación, debido a que la técnica los considera como fallos desconocidos y detiene la ejecución. El caso d si produce

trayectoria de recuperación, tal como es el comportamiento esperado al usar la técnica.

C.3) Simulación de la Inclusión de Nueva Recuperación

En la Fig. 12, se muestran las trayectorias usadas para estimar un tercer modelo TP-GMM, que permite recuperar ante fallos de ejecución al momento de inserción de la manga en el brazo, este fallo se da debido a que el borde de la manga se atrapa con la mano y genera fuerzas en el sensor. En este caso, la recuperación consiste en devolver la manga hasta un nuevo punto previo a la inserción de la manga en el brazo.

En la Tabla II se muestran casos de prueba realizados al incluir el nuevo fallo de ejecución, en cada caso se usan los parámetros de la tarea correspondiente a una trayectoria base o a los valores que resultan de combinar dos de ellas.

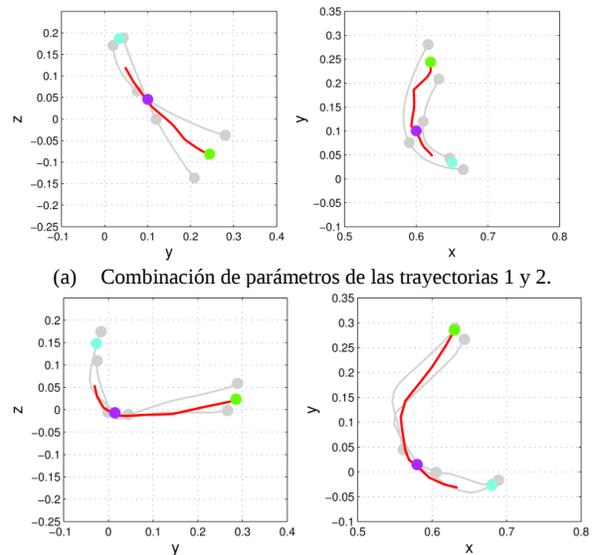
C.4) Discusión

Es de anotar, que en los casos a-c de la Fig. 11, con aumento en la fuerza en el tiempo $t = 7.3s$, no se presentaron durante pruebas reales. Sin embargo, se optó por simularlos, para probar el comportamiento del algoritmo en diversas situaciones.

TABLA II: PRUEBAS REALIZADAS VARIANDO LOS PARÁMETROS DE LA TAREA Y DOS CASOS DE FALLOS.

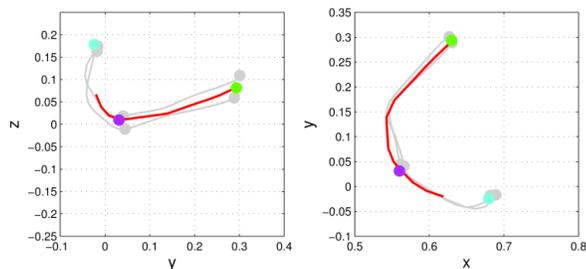
Trayectoria o combinación	Fallo en $t = 3.2s$	Fallo en $t = 7.3s$
1	R	MD
2	R	MD
3	FD	FD
4	FD	FD
5	MD	R
6	MD	R
1 y 2	R	MD
3 y 4	FD	FD
5 y 6	MD	R

R: Recuperación, MD: Modelo desconocido, FD: Fallo Desconocido.

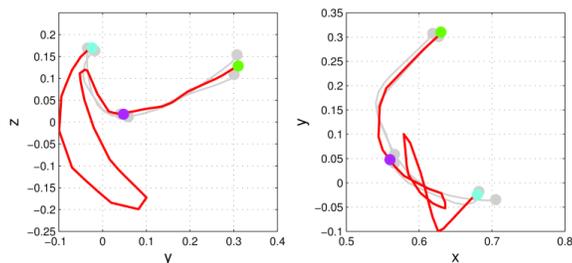


(a) Combinación de parámetros de las trayectorias 1 y 2.

(b) Combinación de parámetros de las trayectorias 3 y 4.



(c) Combinación de parámetros de las trayectorias 4 y 5.



(d) Combinación de parámetros de las trayectorias 5 y 6.

Figura 11: Ejemplo de simulaciones de la ocurrencia de fallo de ejecución en $t = 7,3s$, para nuevos parámetros de la tarea.

Un problema que se puede presentar es la selección de los parámetros de la tarea de un tipo de fallo de ejecución, que sean similares a los de uno existente en la base de datos. Esto ocasionaría una respuesta errónea en la consulta en el árbol-kd.

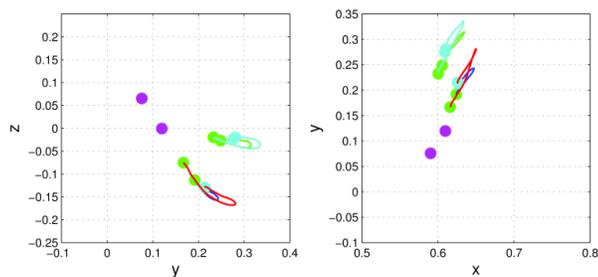


Figura 12: Trayectorias de recuperación de fallo de ejecución de inserción, vistas desde los planos yz y xy . Los círculos en color (verde, azul, violeta) son los valores de b en los parámetros de la tarea.

IV. CONCLUSIONES

En tareas que involucren seres humanos y objetos deformables es necesario emplear técnicas de aprendizaje que sean tolerantes a cambios y a posibles fallos de ejecución. En este trabajo se ha propuesto tratar lo anterior usando una combinación de: i) un árbol-kd, ii) modelos de mezcla de gaussianas parametrizado en la tarea. El sistema implementado permite que el robot coloque la manga de la camisa para diversas posiciones y curvas del brazo y además, pueda recuperarse ante un fallo en la ejecución de la tarea. La trayectoria directa se ejecuta correctamente la mayoría de las veces, pero en algunas ocasiones se pueden presentar enganches de la manga con el brazo, los cuales son detectados por un sensor de fuerza/torque. Es en este punto donde se ejecuta un modelo alternativo, el cual recupera el fallo de ejecución, logrando la colocación de la manga.

Tanto la trayectoria directa como las que recuperan, fueron obtenidas usando programación por demostración. El problema del enganche al colocar la manga, se logró corregir en los dos casos más frecuentes: enganche cerca al hombro y enganche de inserción.

AGRADECIMIENTOS

Este trabajo fue financiado por la Universidad Nacional de Colombia y la Universidad del Quindío, y parcialmente por los proyectos Robinstruct (MINECO TIN2014-58178-R) y I-DRESS (ERA-net CHISTERA PCIN-2015-147). J. Hoyos agradece además al Institut de Robòtica i Informàtica Industrial CSIC/UPC en Barcelona por su colaboración.

REFERENCIAS

- [1] Aude Billard, Calinon Sylvain, Ruediger Dillmann, and Schaal Stefan, "Robot Programming by Demonstration," in *Handbook of Robotics*, Oussama Khatib Bruno Siciliano, Ed. Berlin: Springer, 2008, ch. 59, pp. 1371-1394.
- [2] Rudiger Dillmann et al., "Advances in Robot Programming by Demonstration," *KI - Künstliche Intelligenz*, vol. 24, no. 4, pp. 295-303, 2010. [Online]. <http://dx.doi.org/10.1007/s13218-010-0060-0>
- [3] A. Ude, A. Gams, T. Asfour, and Jun Morimoto, "Task-Specific Generalization of Discrete and Periodic Dynamic Movement Primitives," *Robotics, IEEE Transactions on*, vol. 26, no. 5, pp. 800-815, 2010.
- [4] Sylvain Calinon, Zhibin Li and Tohid Alizadeh and Nikos G. Tsagarakis, and Darwin Caldwell, "Statistical dynamical systems for skills acquisition in humanoids," in *8th IEEE-RAS International Conference on Humanoid Robots*, 2012, pp. 323-329.
- [5] G. Chang and Dana Kulic, "Robot Task Error Recovery Using Petri Nets Learned from Demonstration," in *International Conference on Advanced Robotics (ICAR)*, 2013, pp. 1-6.
- [6] C. Merikli and M. Veloso, "Biped Walk Learning Through Playback and Corrective Demonstration," in *Proceedings of the Twenty-Fourth Conference on Artificial Intelligence (AAAI'10)*, 2010, pp. 1-6.
- [7] L. Rozo, S. Calinon, D. G. Caldwell, P. Jimenez, and C. Torras, "Learning collaborative impedance-based robot behaviors," in *Conference on Artificial Intelligence (AAAI)*, 2013, pp. 1422-1428.
- [8] D. Shinohara, T. Matsubara, and M. Kidode, "Learning motor skills with non-rigid materials by reinforcement learning," in *IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 2011, pp. 2676-2681.
- [9] T. Tamei, T. Matsubara, A. Rai, and T. Shibata, "Reinforcement learning of clothing assistance with a dual-arm robot," in *11th IEEE-RAS International Conference on Humanoid Robots*, 2011, pp. 733-738.
- [10] Jon Louis Bentley, "Multidimensional Binary Search Trees Used for Associative Searching," *Commun. ACM*, vol. 18, no. 9, pp. 509-517, sep 1975.
- [11] Sylvain Calinon, Tohid Alizadeh, and Darwin Caldwell, "On Improving the Extrapolation Capability of Task-Parameterized Movement Models," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2013, pp. 610-616.
- [12] A. Ramisa, G. Aleny, F. Moreno-Noguer, and C. Torras, "Using depth and appearance features for informed robot grasping of highly wrinkled clothes," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2012, pp. 1703-1708.
- [13] S. Calinon, F. Dhalluin, E. Sauser, D. Caldwell, and A. Billard, "Learning and Reproduction of Gestures by Imitation," in *IEEE robotics and automation magazine*, 2010, pp. 44-54.