

## Diseño de un Controlador Híbrido en Ambientes Virtuales para Teleoperación Robótica

Cecilia García\*, Roque Saltaren\*, Ricardo Carelli\*\*\*, Maria Artigas\*\* y Rafael Aracil\*

\*Centro de Automática y Robótica UPM – CSIC

Universidad Politécnica de Madrid, C/ José Gutiérrez Abascal, nº2, 28006,  
Madrid, España (e-mail: cecilia.garcia@upm.es)

\*\* Departamento de Ingeniería Eléctrica. Universidad del Zulia, 4001 Venezuela  
(e-mail: artgasmaria@gmail.com)

**Resumen:** En este artículo se presenta el diseño e implementación de un controlador híbrido para un sistema de teleoperación robótica. La estructura de control de bajo nivel fue diseñada para controlar tanto la posición del manipulador remoto como así también la fuerza de interacción. En la estación local se consideró la dinámica del operador humano y en el sitio remoto se consideró un modelo elástico para el entorno. El controlador supervisor es diseñado utilizando la teoría de sistemas híbridos y en particular se empleó un autómata como herramienta de modelado para el controlador de eventos discretos. Los resultados teóricos se validan a través de una plataforma de simulación de un robot comercial. Copyright © 2010 CEA.

**Palabras Clave:** Telerrobótica, Control de Robots, Sistemas de Eventos Discretos, Simulación, Diseño de Sistemas de Control).

### 1. INTRODUCCIÓN

El control supervisor es una estrategia de control que se emplea ampliamente en el área industrial, militar, médica, etc. (Padala *et al.*, 2007, Jinbo *et al.*, 2004, Seidl 2006, Lin 2004, Caltenco-Arciniega *et al.*, 2006, Chen and Wonham, 2001). Sin embargo, en la mayor parte de los sistemas de teleoperación de robots el control supervisor lo ejerce el operador humano a través de los comandos que envía al sitio remoto (Sheridan 1992). Por lo tanto, al sistema local debe llegar información del modelo del proceso, la tarea a desarrollar, el entorno, etc, para que el operador tome las decisiones adecuadas. Algunas de las motivaciones con las que se cuenta para investigar esta modalidad de control son:

- Mejorar la precisión y la fiabilidad del sistema.
- Hacer que el control sea más rápido y que no se vea restringido por la capacidad sensorial-motora del operador humano.
- Facilitar el control haciendo que el operador dé instrucciones en términos de objetos a mover y objetivos a reunir en lugar de indicar qué instrumentos se deben usar y que señales deben ser enviadas.
- Eliminar la demanda de atención continua por parte del operador y reducir la cantidad de tarea que debe ejecutar.
- Hacer posible el control del sistema aún cuando exista retardo de tiempo entre ambas estaciones de trabajo.

Cuando en un sistema de teleoperación existe retardo en la comunicación entre la estación local y la remota, el control supervisor aparece como una alternativa para solucionar los conflictos provocados por esta característica inherente del sistema (Sheridan 1992, Çavuşoglu, 1997). Tal como se sugiere en la literatura, la implementación de la estrategia de control supervisor requiere un mínimo conocimiento del sistema remoto y de la tarea a desarrollar para que el operador pueda evaluar lo que sucede en la estación remota, predecir el posible

comportamiento del sistema para luego tomar alguna decisión al respecto.

En este trabajo se presenta el diseño e implementación en una plataforma de simulación de un controlador supervisor en la estación remota de trabajo modelado usando la teoría de autómatas. La ventaja clara que se obtiene al ubicar al controlador supervisor en la estación remota es que ésta se convierte en un sistema autónomo pudiendo corregir posibles errores, continuando o abortando una tarea ante la interrupción inesperada de la comunicación o bien ante una variación del retardo convirtiéndose así en un efectivo mecanismo de seguridad para el sistema, la tarea y el medio.

Este artículo está organizado de la siguiente manera. En la Sección 2 se presenta el modelo dinámico del sistema de teleoperación. Además se describe brevemente la estrategia de control utilizada para compensar el retardo de tiempo. Esta estrategia está basada en modelo. La Sección 3 describe en detalle el diseño teórico de un supervisor para el sistema de teleoperación dado en la sección previa. La descripción de la plataforma en la cual se realizó la implementación práctica y los resultados obtenidos son presentados en la Sección 4. Finalmente se abordan las conclusiones y futuros trabajos.

### 2. MODELADO DINAMICO DEL SISTEMA TELEOPERADO

En la estación local del sistema de teleoperación se modela al operador humano, al manipulador local y además se dispone de una réplica de la estación remota, la cual forma parte de la estructura de compensación propuesta para compensar el retardo tiempo de comunicación (García *et al.*, 2000a). En Sutton (1990) se presentan varios modelos del operador humano para diferentes aplicaciones. El modelo utilizado en este trabajo es lineal y está definido en (1);

$$G_h(s) = \frac{K_h e^{-\tau_e s} (1 + T_L s)}{(1 + T_N s)(1 + T_I s)} \quad (1)$$

donde  $K_h$  es la ganancia constante del operador,  $\tau_e$  es el retardo de reacción del operador,  $T_L$  es la constante de adelanto,  $T_N$  es la constante neuromuscular, y  $T_I$  es la constante de atraso. Las constantes de atraso y adelanto están vinculadas a la capacidad de adaptación del operador humano a diferentes dinámicas. El modelo lineal del manipulador local con su controlador PD es,

$$x = \frac{k}{(m_l + m_h)s^2 + (b_l + k_v)s + k_p} f_{ef} \quad (2)$$

donde  $k$  es la ganancia del sistema manipulador local – controlador local,  $f_{ef}$  es la fuerza total que se ejerce sobre el manipulador local,  $m_h$  es la inercia del brazo del operador humano,  $k_v$  y  $k_p$  son las ganancias del controlador PD,  $m_l$  y  $b_l$  son los parámetros de inercia y fricción del manipulador local, y  $x$  es el desplazamiento. La réplica de la estación remota en la estación local mejora el desempeño del sistema evitando sobrepicos en la respuesta del mismo. La Figura 1 presenta un esquema de la estación local;

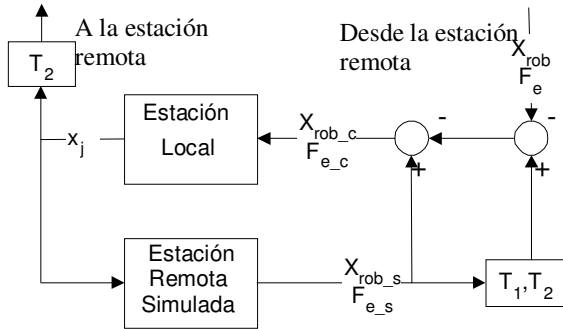


Figura 1: Esquema de la Estación Local.

En la Figura 1,  $x_j$  es la señal de referencia que el operador humano envía a la estación remota,  $x_{rob}$  y  $F_e$  son las señales de posición y fuerza de interacción que son realimentadas desde la estación remota hacia la estación local,  $x_{rob_s}$  y  $F_{e_s}$  son las estimas de las señales  $x_{rob}$  y  $F_e$ , respectivamente,  $x_{rob_c}$  y  $F_{e_c}$  son las señales compensadas enviadas al operador.  $T_1$  y  $T_2$  son los retardos de tiempo que sufren los datos y las imágenes visuales, respectivamente. El modelo de la estación remota simulada es lineal y, dado que se supone conocido el modelo del manipulador remoto de dos grados de libertad, se utilizó un controlador PD con dinámica inversa y lazo de impedancia para su control. Tal como sucede con la estación local, la estación remota tiene dos componentes principales: el manipulador robótico real y el medio en el cual se está realizando la tarea. Además se tiene un modelo de la estación local cuya función es la de predecir las señales de referencia que llegarán desde la estación local. En la ec.(3) se presenta el modelo dinámico del manipulador robótico en interacción con el medio y en coordenadas cartesianas;

$$\mathbf{F} = \mathbf{M}^*(\mathbf{x})\ddot{\mathbf{x}} + \mathbf{C}^*(\mathbf{x}, \dot{\mathbf{x}})\dot{\mathbf{x}} + \mathbf{g}^*(\mathbf{x}) + \boldsymbol{\varphi}^*(\mathbf{x}, \dot{\mathbf{x}}) + \mathbf{f}_e \quad (3)$$

donde  $\mathbf{M}^*(\mathbf{x}) \in \mathbb{R}^{n \times n}$  es la matriz de inercia del manipulador, con  $\mathbf{M}^*(\mathbf{x}) = \mathbf{M}^{*T}(\mathbf{x})$  y  $\mathbf{M}^*(\mathbf{x}) > 0$ ;  $\mathbf{C}^*(\mathbf{x}, \dot{\mathbf{x}}) \in \mathbb{R}^{n \times n}$  es la matriz de fuerzas centrífugas y de Coriolis;  $\mathbf{g}^*(\mathbf{x}) \in \mathbb{R}^n$  es el vector de fuerzas o pares gravitacionales,  $\boldsymbol{\varphi}^*(\mathbf{x}, \dot{\mathbf{x}})$  es el vector

de las fuerzas de fricción,  $\mathbf{x} \in \mathbb{R}^n$  es el vector de posiciones del extremo operativo del robot y  $\mathbf{f}_e$  es la fuerza de interacción entre el extremo operativo del robot y el medio. La ley de control implementada es un control robusto de impedancia (Postigo, 1994) para mejorar el desempeño del sistema de teleoperación ante la presencia de incertidumbres en los parámetros dinámicos del manipulador. En la ec.(4) se presenta dicha ley;

$$\mathbf{F} = \hat{\mathbf{M}}\mathbf{a} + \hat{\mathbf{C}}[\dot{\mathbf{x}} + \nu] + \hat{\mathbf{g}} + \hat{\boldsymbol{\varphi}} + \mathbf{f}_e + \mathbf{K} \text{sgn} \left( \nu^T (\hat{\mathbf{M}}\mathbf{a} + \hat{\mathbf{C}}\nu + \hat{\mathbf{g}} + \hat{\boldsymbol{\varphi}}) \right) \quad (4)$$

donde  $\text{sgn}(\bullet)$  es la función signo y;

$$\mathbf{a} = \ddot{\mathbf{x}}_0 + \mathbf{K}_v(\dot{\mathbf{x}}_0 - \dot{\mathbf{x}}) + \mathbf{K}_p(\mathbf{x}_0 - \mathbf{x}) \quad (5)$$

$$\mathbf{x}_0 = \mathbf{x}_{ref} - \mathbf{x}_a \quad (6)$$

$$\dot{\nu} + \lambda \nu = (\ddot{\mathbf{x}}_0 - \ddot{\mathbf{x}}) + \mathbf{K}_v(\dot{\mathbf{x}}_0 - \dot{\mathbf{x}}) + \mathbf{K}_p(\mathbf{x}_0 - \mathbf{x}) \quad (7)$$

donde  $\hat{\mathbf{M}}$ ,  $\hat{\mathbf{C}}$ ,  $\hat{\mathbf{g}}$  y  $\hat{\boldsymbol{\varphi}}$  son las estimas de los valores reales y tienen la misma estructura que  $\mathbf{M}$ ,  $\mathbf{C}$ ,  $\mathbf{g}$  y  $\boldsymbol{\varphi}$  respectivamente.

$\mathbf{K}_v$  y  $\mathbf{K}_p$  son las matrices de ganancia definidas positivas.  $\mathbf{K}$  es una matriz de ganancia diagonal.  $\mathbf{x}_a$  es la corrección que se le hace a la posición debida a la interacción con el medio. Esta corrección se basa en el concepto de impedancia (Hogan, 1985).  $\mathbf{x}_{ref}$  es la referencia de posición enviada por el operador humano a la estación remota y  $\mathbf{x}$  es la posición actual del robot. En (Sukhan, 1993) se presentó un modelo lineal completo de un sistema de teleoperación que también considera al operador humano y al medio con el cual interactúa el robot remoto. En la Figura (2) se presenta un esquema de la estación remota.

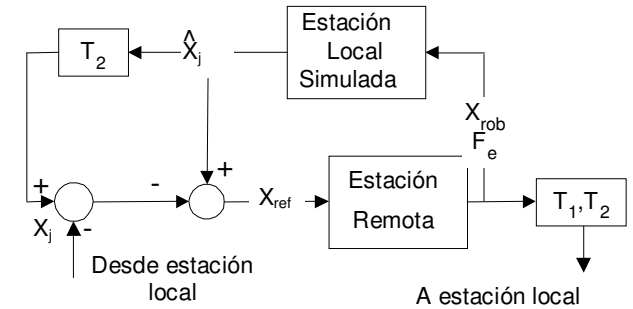


Figura 2: Esquema de la Estación Remota.

La referencia compensada que recibe el robot remoto está dada por;

$$\mathbf{x}_{ref\_c} = \mathbf{x}_{ref\_s} - (\mathbf{x}_{ref\_s} e^{-sT_2} - \mathbf{x}_{ref} e^{-sT_2}) \quad (8)$$

Las señales de fuerza y posición que son realimentadas desde la estación remota a la local son también compensadas y se presentan en las ecs. (9) y (10) respectivamente;

$$\mathbf{x}_{rob\_c} = \mathbf{x}_{rob\_s} - (\mathbf{x}_{rob\_s} e^{-sT_2} - \mathbf{x}_{rob} e^{-sT_2}) \quad (9)$$

$$\mathbf{f}_{e\_c} = \mathbf{f}_{e\_s} - (\mathbf{f}_{e\_s} e^{-sT_1} - \mathbf{f}_e e^{-sT_1}) \quad (10)$$

En los modelos simulados se contemplaron incertidumbres en los parámetros. Con una incertidumbre del 15% y 10% en los parámetros del robot simulado y del medio simulado los resultados de simulación obtenidos muestran un desempeño aceptable (García *et al.*, 2000a). Como puede deducirse de lo anterior, la estructura de compensación está basada en el modelo del sistema por lo que el desempeño de la misma dependerá de cuán exacto sea el conocimiento que se tiene sobre el mismo.

Para disminuir tales efectos y con el fin de darle mayor autonomía al sistema remoto además de aumentar el desempeño del mismo, se diseñó un controlador supervisor basado en la teoría de sistemas híbridos que se presentará a continuación.

### 3. PRELIMINARES

Debido a los avances tecnológicos en la teoría de las comunicaciones, mucho de los sistemas continuos están siendo manejados por eventos. Según Sheridan, (1992), un sistema de eventos discretos está constituido por entidades o procesos que intercambian mensajes o coordinan la ejecución de tareas. Cuando un sistema incluye características propias de un sistema de eventos discretos como así también de un sistema continuo, se obtiene lo que se denomina Sistema Híbrido. Un sistema continuo tiene un espacio de estado continuo y su evolución es modelada a través del paradigma de las ecuaciones diferenciales o ecuaciones en diferencias. Para modelar un sistema de eventos discretos se pueden utilizar varias herramientas como por ejemplo las Redes de Petri, la teoría de Markov, la teoría de Automatas o el álgebra Max+ (Wonham 2003). En particular, en este artículo se modeló al sistema de eventos discretos usando un autómata. Un autómata es un grafo cuyo vértice representa el estado del sistema de eventos discretos y el arco que conecta dos estados es lo que se denomina evento y son los que provocan la evolución del sistema. Un sistema híbrido debe combinar ambas metodología de modelado. En Stiver et al., (1992) se presentó un esquema de un sistema híbrido. El mismo consta de tres partes y en la Figura 3 se reproduce dicho esquema,

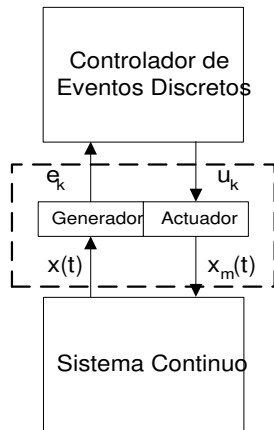


Figura 3: Esquema de un Sistema Híbrido

El sistema continuo, comúnmente denominado planta, tiene su propio espacio de estados continuo y real valuado. Como ya se dijo, la evolución del sistema puede representarse por una ecuación diferencial como la siguiente;

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)) \quad (11)$$

donde  $\mathbf{x}(t) \in \mathbf{R}^n$  es la variable de estado,  $\mathbf{u}(t) \in \mathbf{R}^m$  representa la acción de control y  $\mathbf{f}: \mathbf{R}^n \times \mathbf{R}^m \rightarrow \mathbf{R}^n$  es una función continua que satisface las condiciones de Lipschitz. El controlador de eventos discretos es un sistema discreto que puede ser modelado a través de un autómata determinístico. Típicamente, un autómata determinístico queda definido por una 5-upla como la siguiente:

$$A = \{W, X_d, \tilde{U}, \delta, \Phi\} \quad (12)$$

donde  $W$  es el alfabeto reconocido por el autómata,  $X_d$  es el estado discreto,  $\tilde{U}$  es el conjunto de símbolos generados por el

controlador,  $\delta: X_d \times W \rightarrow X_d$  es la función de transición de estados y  $\Phi: X_d \rightarrow U$  es la función de salida (Antsaklis, 1993).

El controlador y la planta no pueden comunicarse directamente debido a la naturaleza de las señales que cada uno de ellos manejan. Por tal motivo, es necesaria la existencia de una interface que convierta las señales continuas en evento y los símbolos generados por el controlador de eventos discretos en señales continuas.

Tal como ya se mencionara, el control híbrido o supervisor es una de las estrategias que se puede incorporar en un sistema de teleoperación. Este controlador es el encargado de modificar la entrada de referencia de la planta continua cuando se producen interrupciones en la comunicación o bien cuando determinados umbrales de fuerza y posición son superados permitiendo así, coordinar y ejecutar acciones de control para mejorar el desempeño del sistema, particularmente cuando se tienen retardos de tiempo en la comunicación. De este modo, el controlador de eventos discretos, junto con la interface, debe analizar las consignas que el operador humano envía desde la estación local, predecir la acción del manipulador y en caso de que esta predicción no sea la acción deseada. El controlador también debe ordenar una modificación de la referencia para que el manipulador efectivamente se desempeñe según lo deseado. Por otro lado, el controlador de eventos discretos debe detectar una posible interrupción en la comunicación y ordenar una acción de control que posibilite el movimiento del manipulador para continuar la tarea o bien para que el mismo regrese a una posición segura. En la Figura 4 puede observarse la modificación propuesta respecto de la estructura convencional de un sistema de teleoperación cuando se le incorpora la estrategia de control supervisor (Garcia et al., 2000b). La señal de referencia  $\mathbf{x}_{ref}(t)$

que el operador envía desde la estación local a la remota será supervisada por el controlador de eventos discretos antes de ser enviada al robot. El generador se encarga de verificar si estos comandos son correctos de acuerdo con los datos que se dispongan sobre la tarea a realizar o del medio para el diseño del supervisor. En el caso de que la señal enviada por el operador no sea correcta, el generador producirá un evento  $e_k$  provocando el cambio de estado del sistema de eventos discreto y dando con resultado un símbolo  $u_k$ . El actuador interpretará este símbolo y asignará la correspondiente acción continua  $x_m(t)$ .

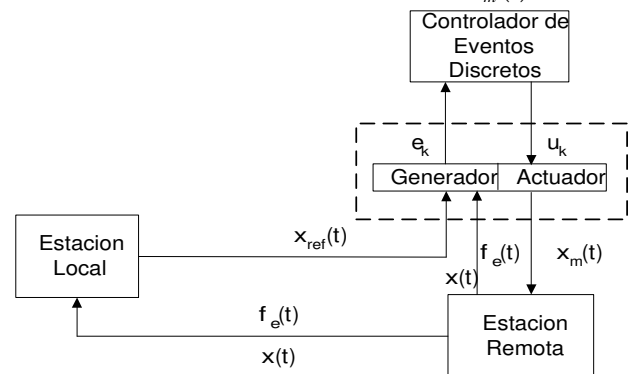


Figura 4: Esquema del Control Supervisor del Sistema de Teleoperación

Antes de presentar formalmente el supervisor diseñado, es necesario ejemplificar la programación de una tarea típica con el objetivo de poder realizar las simulaciones.

Supóngase que se desea realizar una típica tarea en teleoperación

como es la manipulación de un objeto. Es claro que esta tarea se puede dividir en tareas más sencillas como: aproximación al objeto, coger el objeto, mover el objeto, dejar el objeto, volver a situación de reposo. Es decir, es posible realizar la partición del espacio de estados continuo  $\mathbf{X}_{ST}$ , de forma tal que,

$$\mathbf{X}_{ST} = \bigcup_{i=1}^p \mathbf{X}_{STi} \quad (13)$$

donde  $p$  es el número de subtareas a realizar y  $\mathbf{X}_{STi}$  es el espacio de estados asociado a la subtaska STi. Cada partición está separada de la otra por una hipersuperficie que se definirá a continuación. Considerando los valores característicos de cada subtaska, es posible definir un funcional suave cuyo espacio nulo genera una hipersuperficie. Se desea que el supervisor sólo actúe en situaciones que no son previstas por el sistema de control de bajo nivel. Tales situaciones deberían ser identificadas por el generador de eventos. Para el sistema de teleoperación propuesto las situaciones que se desean supervisar se resumen a continuación:

1. Umbrales de posición y/o fuerza.
2. Interrupción de la comunicación.
3. Retorno de la comunicación.
4. Tiempo máximo de espera luego de una interrupción en la comunicación.

Es claro que las acciones a tomar por el controlador de eventos discretos dependen fundamentalmente de la subtaska que se está realizando, es decir que ante una inesperada interrupción de la comunicación, las acciones que el supervisor tomará dependerán de si el robot remoto está en movimiento libre o movimiento restringido. Por lo tanto es conveniente generar eventos que permitan identificar en qué fase de la tarea el sistema de teleoperación está trabajando.

### 3.1 Generador de Eventos Discretos

Debe quedar claro que los eventos que se desean generar dependerán fundamentalmente de la tarea a desempeñar, es por ello que a priori deben conocerse algunos parámetros de la misma como por ejemplo la fuerza máxima que se debe realizar durante la ejecución de la misma. A tales efectos se definirán los eventos que permitirán la evolución del sistema de eventos discretos. Cada evento se identifica por medio de etiquetas que se presentarán a continuación. Estas etiquetas están vinculadas a las señales de fuerza y posición del sistema como así también al estado de funcionamiento de la red de comunicaciones.

Nótese que el diseño de un controlador supervisor ya sea para teleoperación o cualquier otra aplicación está inherentemente vinculado con el sistema en particular, la tarea a desarrollar y las señales a supervisar.

Sea  $I$  un conjunto de etiquetas no vacío definido en (14):

$$I = \tilde{I} \cup \tilde{N} \quad (14)$$

donde  $\tilde{I} = \{PN, PS, FS, FN, ms, cc, rc\}$  y  $\tilde{N} = \{\tilde{n}_i\}_{i=1, \dots, p}$ . Además  $cc$ : comunicación interrumpida,  $rc$ : retorno de la comunicación,  $ms$ : máximo tiempo de espera,  $PS$ : máximo umbral de posición superado,  $PN$ : posición correcta,  $FS$ : máximo umbral de fuerza superado,  $FN$ : fuerza correcta, y el símbolo  $\tilde{n}_i$ : indica que la teleoperación está siendo realizada sin la ocurrencia de eventos durante la subtaska STi. Durante la ejecución de una subtaska, el robot remoto sigue a la referencia dada para la subtaska STi, de forma tal que es posible identificar, a través de un funcional, en cuál de las regiones del

espacio continuo está operando el sistema. Luego, la hipersuperficie que separa a cada una de las regiones está dada por el espacio nulo del funcional. Sea el funcional;

$$f: \mathbf{R}^n \rightarrow \mathbf{R} \quad (15)$$

Cada vez que la planta continua cruce alguna de estas hipersuperficies definidas en la ec.(15), un símbolo  $e_i$  será generado, esto es;

$$e_i = \vartheta(\mathbf{x}_c) \quad (16)$$

donde  $\mathbf{x}_c$  es el valor del estado cuando cruza la hipersuperficie.

$e_i$  es el evento generado cuando el sistema cruza la hipersuperficie  $i$ .  $\vartheta: \mathbf{R} \rightarrow \Gamma$  es la función generadora de eventos y  $\Gamma = \{e_i\}_{i=1, \dots, p}$ . En un sistema de teleoperación,

además de los eventos que se suceden debido al estado continuo de la planta, debe considerarse el estado de la red de comunicación ya que puede funcionar normalmente o puede sufrir interrupciones. Es posible, entonces generar los siguientes eventos;

$$\eta(t_x, d, d_m, \mathbf{x}(t)) \quad (17)$$

donde  $t_x$  es el instante en el cual el dato  $d$  llega a la estación remota,  $d_m$  es el valor del retardo de tiempo el cual se considera

fijo,  $\eta: \mathbf{R} \times \mathbf{R} \times \mathbf{R} \times \mathbf{R}^n \rightarrow \tilde{\eta}$  es el máximo tiempo de espera para el retorno de la comunicación.  $\mathbf{Im}(\tilde{\eta})$  es la función generadora de eventos. Cada elemento del conjunto  $\tilde{\eta} = \{\tilde{\eta}_{cc}, \tilde{\eta}_{rc}, \tilde{\eta}_{ms}\}$  es un evento que indica el estado de la comunicación.

*Definición 1:* Cuando sucede una interrupción en la comunicación, ningún dato llega a la estación remota, por lo tanto el estado bajo esta situación se define como  $x(t) = \emptyset$ .

En un sistema de teleoperación con retardo de tiempo usualmente se dispone de una estrategia de control que permita predecir el comportamiento del manipulador, en el caso de que este no fuera el deseado, se debe poder aplicar una acción correctiva antes de que el manipulador continúe erróneamente ejecutando la tarea. Este tipo de control ayuda a mejorar el desempeño del sistema de teleoperación puesto que el operador humano puede enviar un comando erróneo y deberá esperar  $d_1 + d_2$  segundos para poder enviar la acción correctiva. Para algunas aplicaciones este comportamiento es inadmisibles. Consecuentemente, se desea tener un conjunto de funcionales y funciones que supervisen el estado de la planta o la evolución de algunas señales que resultan de particular interés para el sistema. Por lo tanto se deberá generar un evento si el estado del sistema o la señal en cuestión toma valores no apropiados. En este caso se generará un funcional dado por;

$$v = \xi(\mathbf{h}(t)) \quad (18)$$

donde  $\xi: \mathbf{R} \rightarrow \Omega$  es la función generadora de eventos,  $\Omega$  es el conjunto de eventos y  $v \in \Omega$  es el evento generado cuando el argumento de la función supera el máximo permitido para la subtaska. Con  $\mathbf{h}(t)$  se representa a la variable de estado o a la fuerza de interacción. Para generar dichos eventos se realiza una partición del espacio de estados y cada región es separada de otra a través de una hipersuperficie, de forma tal que cuando el estado cruce dicha hipersuperficie, el generador genera el

evento. Antes de definir formalmente las funciones generadoras de eventos, se presenta la nomenclatura que se utilizará:

- El conjunto de valores máximos de la subtaska ST<sub>i</sub> será denotado como  $X_{M_{STi}}$ .
- El conjunto de valores iniciales de la subtaska ST<sub>i</sub> se denotará por  $X_{O_{STi}}$ .
- El conjunto de valores máximos permitidos para la fuerza de interacción durante la etapa de movimiento restringido de la subtaska  $q$  será denotado como  $F_{M_{STq}}$ .
- Finalmente  $F_{M_{STq}}$ ,  $f_{STi}$ ,  $f_{SVi}$ , y  $f_{SFi}$  son los funcionales generadores de eventos mientras que la tarea ST<sub>i</sub> está siendo realizada.

**Definición 2:** Para la estación remota considerada y para la tarea planificada, el funcional acotado  $f_{STi}$  está definido por,

$$f_{STi} = \left( \left\| \mathbf{x}_{m_{STi}} - \mathbf{x}_{0_i} \right\|_2^2 - \left\| \mathbf{x} - \mathbf{x}_{0_i} \right\|_2^2 \right) e^{-\|f_e\|_\infty} \quad (19)$$

donde  $\mathbf{x}_{m_{STi}} \in X_{M_{STi}}$  y  $\mathbf{x}_{0_i} \in X_{O_{STi}}$ . Sea NST<sub>i</sub> el espacio nulo de  $f_{STi}$ , los eventos asociados a cada subtaska son definidos por la función  $\vartheta(\mathbf{x}_{ci})$ , que se presenta a continuación,

$$\vartheta(\mathbf{x}_{ci}) = \begin{cases} e_1 & \text{if } \mathbf{x}_{c1} \in N_{ST1} \\ e_2 & \text{if } \mathbf{x}_{c2} \in N_{ST2} \\ e_3 & \text{if } \mathbf{x}_{c3} \in N_{ST3} \\ e_4 & \text{if } \mathbf{x}_{c4} \in N_{ST4} \\ e_5 & \text{if } \mathbf{x}_{c5} \in N_{ST5} \end{cases} \quad (20)$$

**Definición 3:** Para la estación remota considerada y para la tarea planificada, el funcional lineal acotado  $f_{SVi}$  es definido como,

$$f_{SVi} = \left\| \mathbf{x}_{m_{STi}} - \mathbf{x}_{0_{STi}} \right\|_2^2 - \left\| \mathbf{x} - \mathbf{x}_{0_{STi}} \right\|_2^2 \quad (21)$$

y los eventos asociados a la variable de estados están definidas por la función  $\xi(\mathbf{x})$ ,

$$\xi_{\mathbf{x}}(\mathbf{x}) = \begin{cases} \xi_{PS} & \text{if } f_{SVi} < 0 \\ \xi_{PN} & \text{if } f_{SVi} \geq 0 \end{cases} \quad (22)$$

**Definición 4:** Para la estación remota considerada y para la tarea planificada, el funcional lineal acotado  $f_{SFi}$  está definido como,

$$f_{SFi} = \left\| \mathbf{f}_{e_{m_{STi}}} \right\|_2^2 - \left\| \mathbf{f}_e \right\|_2^2 \quad (23)$$

donde  $\mathbf{f}_{e_{m_{STi}}} \in F_{M_{STq}}$ . Los eventos asociados a la variable de estado son definidos por la función  $\xi_{\mathbf{f}_e}(\mathbf{f}_e)$  como sigue;

$$\xi_{\mathbf{f}_e}(\mathbf{f}_e) = \begin{cases} \xi_{FS} & \text{if } f_{SFi} < 0 \\ \xi_{FN} & \text{if } f_{SFi} \geq 0 \end{cases} \quad (24)$$

**Definición 5:** Para la estación remota considerada y para un retardo de tiempo conocido y fijo los eventos generados para el estado de la red de comunicación son definidos por la función  $\eta(T, T_{x_i}, d, \mathbf{x}(t))$  de la siguiente manera;

$$\eta(T, T_{x_i}, d, \mathbf{x}(t)) = \begin{cases} \eta_{kc} & \text{if } T > T_{x_i} + d \wedge \mathbf{x}(T_{x_i} + d) = \emptyset \\ \eta_{rc} & \text{if } T > T_{x_i} + d \wedge \mathbf{x}(T_{x_i} + d) \neq \emptyset \\ \eta_{ms} & \text{if } T > T_{x_i} + d + d_{ms} \wedge \mathbf{x}(T_{x_i} + d) = \emptyset \end{cases} \quad (25)$$

donde  $T_{x_i}$  es el tiempo en el cual el dato  $\mathbf{x}_i$  arribó a la estación remota,  $d$  es el retardo de comunicación,  $d_{ms}$  es el máximo tiempo de espera para el retorno de la comunicación,  $\mathbf{x}$  es el estado.

**Definición 6:** Los eventos  $N = \{n_i\}_{i=1 \dots p}$  indican que el sistema de teleoperación está operando de manera correcta y este estado de funcionamiento se produce cuando  $f_{ST1} > 0$ ,  $f_{STi} \geq 0$ , con  $i = 2 \dots p$  y  $\mathbf{x}(T_{x_i} + d) \neq \emptyset$ .

Sea un conjunto no vacío definido como  $S = \text{Im}(\xi_{\mathbf{x}}) \cup \text{Im}(\xi_{\mathbf{f}}) \cup \text{Im}(\eta) \cup \text{Im}(N)$ , los elementos de  $S$  pueden escribirse como  $(S_{\alpha})_{\alpha \in I}$ .

### 3.2 Controlador de Eventos Discretos

En la ec.(8) se presentó un autómata genérico  $A$ , en lo que sigue se particularizará cada uno de sus componentes. Se asume que el autómata  $A$  es determinístico (es decir que el supervisor selecciona un único símbolo de salida  $u_k$  para cada símbolo de entrada  $x_k$ ).  $A$  es causal (si dos símbolos de entrada  $x_i$  y  $x_j$  son iguales en el instante  $t_i$ , entonces los símbolos de salida  $\tilde{u}_i$  y  $\tilde{u}_k$  son iguales en el instante  $t_i$ ).  $A$  es estrictamente discreto (una secuencia de salida  $\tilde{u}$  depende de una única secuencia de entrada  $x$  y no de inter-eventos) (Antsaklis et al., 1993). La Figura 5 muestra el autómata diseñado. Se puede observar que el diseño del autómata está estrechamente vinculado a la trayectoria programada y se asume que el operador respetará los parámetros característicos de la tarea pre programada como son por ejemplo los valores límites en posición y fuerza que podrá alcanzar. El estado final de cada subtaska se indica como estados marcados. De la Figura 5 se puede deducir la función de transición de estados  $\delta$  y la función de salida  $\Phi$ . Los símbolos generados por el controlador de eventos discretos dependen del evento ocurrido ( $s_{\alpha}$ ) y la subtaska que está siendo realizada ( $e_i$ ), esto es,

$$D: S \times \Gamma \rightarrow \tilde{U} \quad (26)$$

donde  $\tilde{U}$  es el conjunto de símbolos entregada por el controlador al actuador. Un elemento de este mapeo está simbolizado por  $\tilde{u}_j = d(s_j, e_i)$  con  $i = 1 \dots p$  y  $j \in I$ .

### 3.3 Diseño del Actuador

Así como el generador transforma las señales continuas en eventos, el actuador debe interpretar los símbolos dados por el controlador de eventos discretos y debe convertirlos en señales continuas que serán entregadas al robot remoto como referencias. Sea  $X^{sc}$  un conjunto de señales de referencia

que serán dadas al robot remoto. Entonces el mapeo  $C: \tilde{U} \rightarrow X^{sc}$ , se define como:

$\{u_{n_i}\} \rightarrow \{x_{u_i}(t)\}$ , donde  $\{x_{u_i}(t)\}$  es la referencia enviada por el operador humano.

$u_{cc_1} \rightarrow x_{1_{-1}}(t)$  donde  $x_{1_{-1}}(t)$  es la referencia alternativa programada.

$u_{PS_1} \rightarrow x_{2_{-1}}(t)$  donde  $x_{2_{-1}}(t) = \text{step}(x_{m_{ST_1}})$ .

$u_{ms_1} \rightarrow x_{3_{-1}}(t)$  donde  $x_{3_{-1}}(t)$  es la referencia que el robot deberá seguir para alcanzar una posición de seguridad.

$u_{cc_2}, u_{PS_2}, u_{ms_2}$  ídem a  $u_{cc_1}, u_{PS_1}, u_{ms_1}$  para la subtaska  $ST_2$ .

$u_{cc_3} \rightarrow x_{1_{-3}}(t)$  ídem a  $x_{3_{-1}}(t)$  para  $ST_3$ .

$u_{PS_3} \rightarrow x_{2_{-3}}(t)$  donde  $x_{2_{-3}}(t) = \text{step}(f_{m_{ST_3}})$ , pero es mapeada a una posición de referencia pero a través del modelo del medio.

$u_{PS_3} \rightarrow x_{3_{-3}}(t)$  ídem a  $x_{2_{-1}}(t)$  para  $ST_3$ .

$u_{cc_4}, u_{PS_4}, u_{ms_4}$  ídem a  $u_{cc_3}, u_{PS_3}, u_{ms_3}$  para  $ST_4$ .

Cuando el controlador de eventos discretos emite un símbolo  $u_{PN_i}, u_{fn_i}$  o  $u_{rc_i}$ , el actuador debe interpretar que el sistema de teleoperación volvió al modo normal de operación y que la referencia dada al robot remoto la emite el operador.

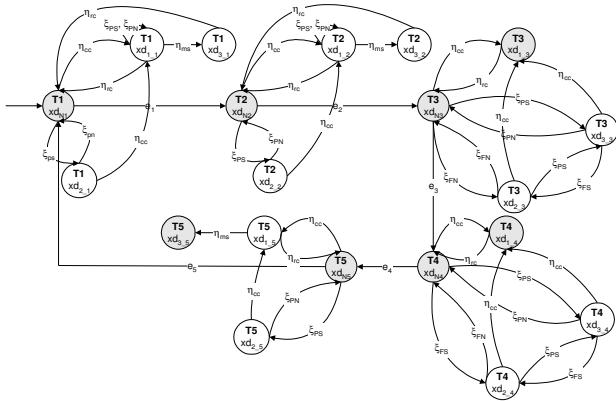


Figura 5: Autómata Diseñado para el Controlador de Eventos Discretos

#### 4. RESULTADOS DE SIMULACIÓN

En esta sección se presentan los resultados de simulación obtenidos al implementar el sistema de teleoperación junto con la estructura de compensación del retardo de tiempo que se presentó brevemente en la Sección 2 y además el supervisor diseñado. Para tal efecto se utilizó Matlab y Simulink y se presentan los resultados obtenidos al considerar eventos vinculados con el estado de la comunicación. A continuación se describe la secuencia de eventos implementados.

Una interrupción en la comunicación se produce a los 4 segundos durante la ejecución de la subtaska  $ST_1$ , luego a los 7 segundos se restablece la comunicación entre ambas estaciones de trabajo. Durante el tiempo que la comunicación estuvo interrumpida, el supervisor generó una referencia alternativa para el robot remoto que permitió continuar con la tarea y la estación remota se convirtió en un sistema autónomo. En la

Figura (6) se presenta la referencia y la trayectoria del sistema en la dirección  $x$  del plano de movimiento. Cuando la comunicación se interrumpe, el supervisor se comporta como si fuera el operador humano puesto que es éste quien está generando la nueva referencia de movimiento para el robot remoto. Es claro que la referencia que genera el supervisor no es la misma que generaría el operador humano, sin embargo el robot remoto sigue esta nueva consigna hasta la ocurrencia de un nuevo evento. Cuando la comunicación retorna, es el operador quien envía los comandos.

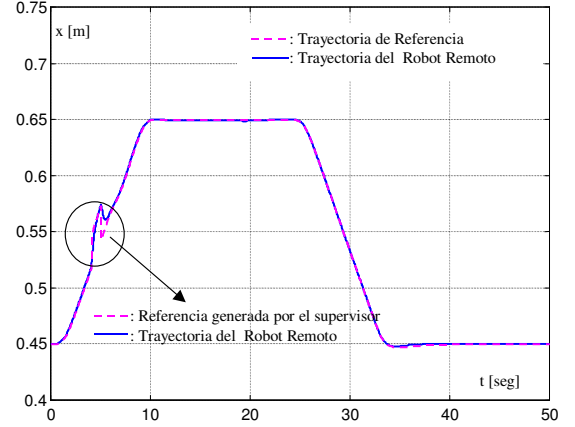


Figura 6: Referencia y trayectoria del Robot remoto en la dirección  $x$ .

En la Figura (7) se presenta el desempeño del sistema en la dirección  $y$  de movimiento. Tal como ocurre en el caso anterior, al interrumpirse la comunicación el supervisor genera una referencia alternativa para que el sistema evolucione también en esta dirección, esta referencia es seguida por el robot hasta la ocurrencia de un nuevo evento.

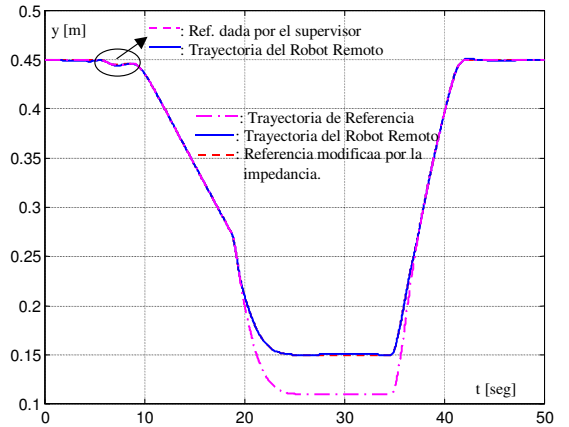


Figura 7: Referencia y Trayectoria del robot remoto en la dirección  $y$ .

En la Figura (8) se presenta una gráfica que permite comparar el desempeño del sistema cuando el mismo evoluciona normalmente, es decir sin la ocurrencia de eventos no controlables, y cuando el supervisor genera referencias para que el robot remoto continúe una tarea. Nótese que en los casos presentados el manipulador remoto está en movimiento libre y es por ello que ante la interrupción de la comunicación es posible darle una referencia alternativa para que continúe con la tarea. Si el robot estuviera en interacción con el medio,

tal vez no sea aconsejable que el sistema remoto sea autónomo. Sin embargo las acciones que tomará el actuador del sistema híbrido dependerán exclusivamente del grado de complejidad de la tarea a desarrollar y del sistema remoto.

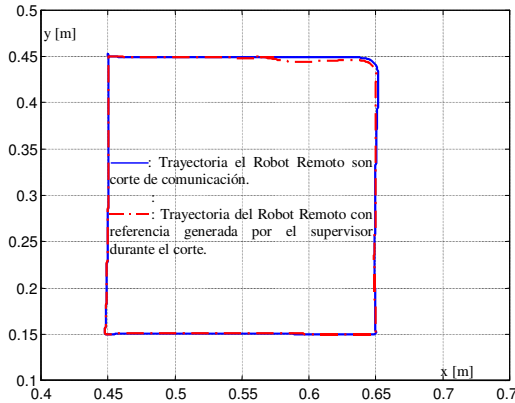


Figura 8: Trayectoria del Robot Remoto en el espacio de trabajo.

## 5. RESULTADOS EXPERIMENTALES

En esta plataforma de simulación se dispone de un joystick con reacción de fuerzas de Microsoft SideWinder Force Feedback y un simulador Robot3D del robot Scara BOSCH SR800 de 4 grados de libertad. Simulador implemente el modelo cinemática y dinámico del robot que fue desarrollado y validado en trabajos previos (Castro, 1999). La aplicación del simulador Robot3D esta desarrollada en Visual C++ 6.0 de Microsoft y utiliza el lenguaje OpenGL para la visualización 3D. Funciona en cualquier plataforma Windows que soporte OpenGL. En la Figura (10) puede observarse una foto de la plataforma de simulación.

Figura 10: Plataforma de Simulación

Esta aplicación permite cargar distintos modelos de robots para su simulación desde archivo que indican su configuración además de tener la capacidad de comunicarse con otros procesos a través memoria compartida, esto posibilita su utilización ya sea con dispositivos de entrada/salida externos (Joystick con reacción de fuerzas) u otra aplicación que actúa como controlador. El modelo identificado e implementado en C++ se presenta a continuación:

Los componentes del vector de parámetros  $[\theta_{vi}]$  con  $i=1, \dots, n$  dependen de una combinación de los parámetros físicos de los eslabones tales como masas, centros de masas, momentos

de inercia y coeficientes de fricción, así como parámetros dinámicos y eléctricos de los motores de actuación los cuales se consideraron invariantes en el tiempo. El controlador implementado es un controlador PD y además se le incorporó un algoritmo de impedancia con el fin de hacer modificar la posición dada por el operador cuando se realiza el contacto con el medio. La impedancia programada es lineal y está dada por  $Z^{-1} = Bs + C$ . La misma se ajustó con los valores  $B=1$  y  $C=10$ .

$$V = \begin{bmatrix} R_1(\theta_{v1} + \beta_1\theta_v + 2\beta_1\theta_{v3}cq_2) & R_1\beta_1(\theta_{v4} + \theta_{v3}cq_2) \\ R_2\beta_2(\theta_{v5} + \theta_{v6}cq_2) & R_2(\theta_{v7} + \beta_2\theta_{v5}) \end{bmatrix} \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{bmatrix} + \begin{bmatrix} R_1(\theta_{v8} - \beta_1\theta_{v3}sq_2\dot{q}_2) & -R_1\beta_1\theta_{v3}sq_2(\dot{q}_1 + \dot{q}_2) \\ R_1\beta_2\theta_{v6}sq_2\dot{q}_1 & R_2\theta_{v9} \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} + \begin{bmatrix} R_1\theta_{v10} \operatorname{sgn}(\dot{q}_1) \\ R_1\theta_{v11} \operatorname{sgn}(\dot{q}_2) \end{bmatrix} \quad (27)$$

donde

$$\begin{aligned} \theta_{v1} &= \gamma_1 J_{m1}, & \theta_{v2} &= \gamma_1 (m_1 r_1^2 + m_2 I_1^2 + m_2 r_2^2 + I_1 + I_2), \\ \theta_{v3} &= \gamma_1 m_2 I_1 r_2, & \theta_{v4} &= \gamma_1 (m_2 r_2^2 + I_2), & \theta_{v5} &= \gamma_2 (m_2 r_2^2 + I_2), \\ \theta_{v6} &= \gamma_2 m_2 I_1 r_2, & \theta_{v7} &= \gamma_2 J_{m2}, & \theta_{v8} &= \gamma_1 \alpha_1, & \theta_{v9} &= \gamma_2 \alpha_2, & \theta_{v10} &= \gamma_1 f_{c1}, \\ \theta_{v11} &= \gamma_2 f_{c2}. \end{aligned}$$

### 5.1 Diseño del Generador de Eventos

A diferencia de lo que se consideró en el desarrollo teórico del supervisor y en las pruebas de simulación realizadas, en la implementación práctica fue necesario considerar a cada una de las articulaciones y no sólo al extremo final. En la ec. (28) se presenta la función  $f_{Cart_i}$  la cual define la hypersuperficie de cruce que determinará la generación de un evento vinculado a la posición cartesiana,

$$f_{Cart_i} = \|\mathbf{x}_{max_i} - \mathbf{x}_{0_i}\|_2^2 - \|\mathbf{x}_i - \mathbf{x}_{0_i}\|_2^2 \quad (28)$$

En este caso con  $\mathbf{x}_{max_i}$  se representa la máxima posición cartesiana que puede alcanzar el extremo de cada una de las articulaciones. En particular puede representarse la posición de un obstáculo en el espacio. Con  $\mathbf{x}_i$  se representa la posición actual del extremo final de cada articulación. Finalmente con  $\mathbf{x}_{0_i}$  se representa la posición inicial de la articulación  $i$  en el espacio cartesiano. La función se evalúa para cada valor de posición articular y luego se genera el evento a partir del valor que toma la función. La función generadora de eventos se presenta en la ec. (29),

$$\xi_x(f_{Cart_i}) = \begin{cases} \xi_{PN} & \text{si } f_{Cart_i} < 0 \\ \xi_{PS} & \text{si } f_{Cart_i} \geq 0 \end{cases} \quad (29)$$

Para la generación de eventos relacionados con la posición articular de cada brazo, se utilizó el hecho de que cada articulación al moverse describe un sector circular. Se comparará el área del sector circular que describe la articulación  $i$  con el máximo área que se le permite describir.



$$f_{Qart_i} = l_i^2(q_{max_i} - q_{min_i}) - l_i^2(q - q_{min_i}) \quad (30)$$

donde  $l_i$  representa la longitud del brazo  $i$ ,  $q_{max_i}$  es el valor máximo que puede alcanzar la articulación  $i$ ,  $q_{min_i}$  es el valor mínimo en coordenadas articulares mientras que  $q$  es la posición articular actual. La generación de eventos se realiza utilizando una función que considere el valor que toma el funcional.

$$\xi_q(f_{Qart_i}) = \begin{cases} \xi_{QS\_U} & \text{si } f_{Qart_i} < 0 \\ \xi_{QS\_L} & \text{si } f_{Qart_i} > l_i^2(q_{max_i} - q_{min_i}) \\ \xi_{QN} & \text{si } 0 \leq f_{Qart_i} \leq l_i^2(q_{max_i} - q_{min_i}) \end{cases} \quad (31)$$

Para generar los eventos vinculados a la señal de fuerza se define el funcional lineal acotado  $f_{SFi}$  como,

$$f_{SFi} = \|\mathbf{f}_{em}\|_2^2 - \|\mathbf{f}_e\|_2^2 \quad (32)$$

donde  $\mathbf{f}_{em}$  es la máxima fuerza de interacción permitida al robot remoto. Los eventos asociados a la variable de estado son definidos por la función  $\xi_f(\mathbf{f}_e)$  como sigue;

$$\xi_f(\mathbf{f}_e) = \begin{cases} \xi_{FS} & \text{if } f_{SFi} < 0 \\ \xi_{FN} & \text{if } f_{SFi} \geq 0 \end{cases} \quad (33)$$

La implementación de cada una de las funciones dadas en las ec. (28) a (33) se realizó utilizando C++. En lo que se refiere al estado de la red de comunicaciones, tanto el corte como el restablecimiento de la comunicación se realizaron por software utilizando las funciones programadas por el programa cliente. Por lo tanto, cuando se produce la interrupción de la comunicación una señal binaria generada por software llega al generador de eventos y éste genera el evento correspondiente. En la ec.(34) se puede observar la sencilla función con la cual se generan los eventos

$$\eta(b) = \begin{cases} \eta_{cc} & \text{si } b = 1 \\ \eta_{rc} & \text{si } b = 0 \end{cases} \quad (34)$$

#### 5.4 Autómata Implementado

El autómata diseñado se presenta en la Figura (11). Los estados del autómata se identificaron con  $ST_i$  donde  $i$  indica la cardinalización del estado en cuestión.

Con  $\alpha_i$  se indica el símbolo generado por el autómata y que luego deberá ser interpretado por el actuador.  $ST_1$  es el estado inicial, en este estado el sistema de teleoperación evoluciona según los comandos que el operador humano envía desde la estación local haciendo uso del manipulador local. Sin embargo el supervisor está verificando que ninguno de los comandos dados superen los valores permitidos.  $ST_2$  y  $ST_6$  son los estados a los que se llega cuando el supervisor detectó que el comando enviado por el operador conducirá a una posición articular no permitida o alcanzable por el manipulador remoto.  $ST_3$  es un estado del autómata al que se puede llegar por dos caminos. Por uno de estos caminos, el estado es alcanzable luego de que el supervisor ha corregido

la posición en coordenadas articulares y el siguiente paso es verificar que la nueva posición que le será dada al robot remoto, no conduzca a una colisión con el medio. El otro camino por el cual puede alcanzarse el estado en cuestión es a través de la ocurrencia de un evento  $\xi_{PS}$ . En este caso las consignas de posición que son enviadas desde la estación local a la remota son consignas de posición en coordenadas cartesianas. Para salir de este estado y regresar al estado  $ST_1$ , el generador debe generar un evento  $\xi_{PN}$ . Este evento ocurrirá sólo en el caso de que la posición cartesiana sea correcta, esto es; que ninguno de los extremos finales de cada articulación no superen los máximos permitidos. Luego de que se haya verificado la posición cartesiana, es necesario verificar nuevamente la posición en coordenadas articulares.

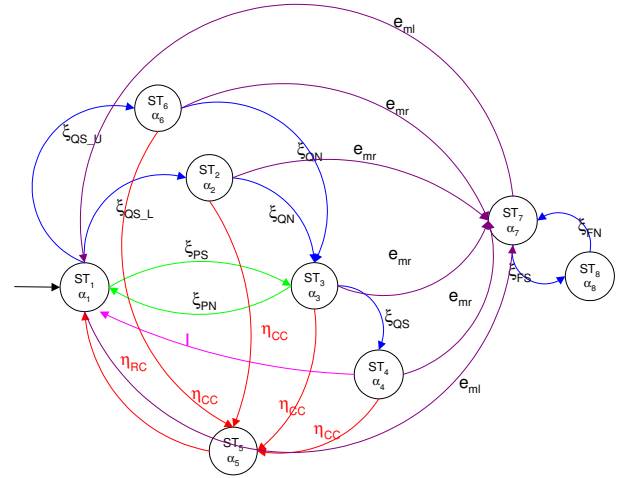


Figura 11: Autómata Implementado

En el caso de que la posición cartesiana corregida por el autómata supere los valores articulares máximos permitidos, el generador dará el evento  $\xi_{QS}$ . En este caso el autómata evoluciona y cambia al estado  $ST_4$ . Ante la ocurrencia de un evento  $\eta_{cc}$  el sistema de eventos discretos cambia de estado, pasando al estado  $ST_5$  en el cual se indica que no hay comunicación entre la estación local y la remota. Los estados  $ST_7$  y  $ST_8$  se alcanzan cuando existe fuerza de interacción. Si la fuerza alcanzada por el extremo final supera los valores máximos permitidos, se generará un evento  $\xi_{FS}$ . Cuando los valores de fuerza sean normales se recupera el estado normal de teleoperación. Cada estado genera un símbolo que es interpretado por el actuador. En la próxima sección se hará una detallada descripción del diseño del actuador.

#### 5.5 Diseño del Actuador

Al diseñar la función que se encargará del mapeo del espacio de símbolos de salida entregados por el autómata al espacio de las señales continuas que deben entregarse al sistema continuo, se consideró como principal condición el hecho de que el sistema que se controlará es complejo desde el punto de vista mecánico. Del mismo modo la tarea de excavación no podría automatizarse completamente debido a la



complejidad de la misma. Por lo que las acciones que tomará el controlador supervisor ante la ocurrencia de eventos son sencillas desde el punto de control. El conjunto de símbolos que genera el autómata está dado en la ec.(35):

$$\tilde{U} = \{\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6, \alpha_7, \alpha_8\} \quad (35)$$

donde  $\alpha_i$  es el símbolo de salida dado por el autómata en el estado discreto  $i$ . El conjunto de las señales continuas al que mapeará el actuador está dado en la ec.(36);

$$X^{SC} = \{X_{max_i}\} \cup \{Q_{max_i}\} \cup \{Q_{min_i}\} \cup \{X_{ref}\} \cup \{F_{max}\} \quad (36)$$

donde  $X_{ref}$  es el conjunto de referencias dadas por el operador humano desde la estación local de trabajo,  $F_{max}$  es la máxima fuerza de interacción que se desea que el extremo del robot aplique al medio. Tanto  $X_{max_i}$  como  $Q_{max_i}$  son los conjuntos cuyos elementos son los valores máximos permitidos en coordenadas cartesianas y articulares respectivamente para el extremo final de la articulación  $i$ . Del mismo modo  $Q_{min_i}$  no es otra cosa que el mínimo valor que puede alcanzar la coordenada articular  $i$ . A continuación, en la ec. (37), se presenta la función diseñada para el actuador:

$$\mathbf{x}_m(t) = \begin{cases} \mathbf{x}_{ref}(t) & \text{si } \alpha_1 \\ q_{min_i}(t) & \text{si } \alpha_2 \\ q_{max_i}(t) & \text{si } \alpha_3 \\ \mathbf{x}_{max}(t) & \text{si } \alpha_4 \\ \mathbf{i} & \text{si } \alpha_5 \\ \mathbf{x}_{ref}(t) & \text{si } \alpha_6 \\ \mathbf{f}_{max}(t) & \text{si } \alpha_7 \\ \mathbf{f}_{ref}(t) & \text{si } \alpha_8 \end{cases} \quad (37)$$

Aquí se presentan los resultados obtenidos al implementar el controlador supervisor en el simulador presentado con anterioridad. La Figura 12 muestra los experimentos realizados con el simulador del Robot3D. En ella se puede observar el desempeño del sistema cuando el operador humano realiza la teleoperación sin el controlador supervisor. Nótese que el extremo final del manipulador remoto colisiona con una pared que fue simulada dentro del espacio de trabajo. En la simulación, parte del brazo robótico se introduce en la pared pasando a través de ella. Sin embargo, en una situación real el sistema remoto puede incurrir en un movimiento no deseado provocando daños en el extremo operativo del brazo o bien en el entorno de trabajo. Cuando el supervisor es implementado, el comportamiento del sistema de teleoperación es diferente y el desempeño del sistema remoto se ve alterado favorablemente por la acción del supervisor, evitando situaciones peligrosas como la descrita anteriormente.

En la Figura 13 se presenta la fuerza que el operador humano siente en su mano (fuerza de interacción reflejada). Nótese que no sólo se percibe el contacto con el objeto, sino que también se siente la dinámica del robot que está teleoperando. En las Figuras 14 y 15 se presenta el accionar del supervisor

en fuerza (direcciones  $x$  e  $y$  del espacio). Nótese que cuando el manipulador comienza a interactuar con el medio, de no existir el supervisor se ocasionarían grandes daños tanto en la herramienta del robot remoto, como así también en el medio.

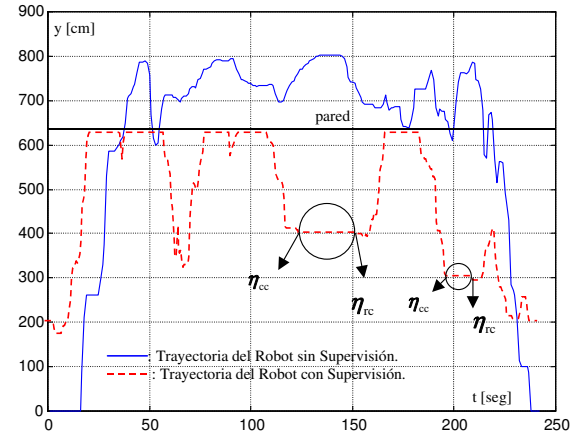


Fig. 12. Supervisión de la posición del robot remoto

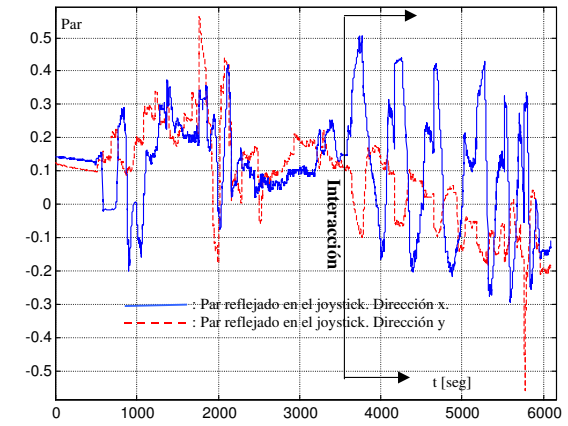


Fig. 13: Reflexión de Fuerza en el Joystick

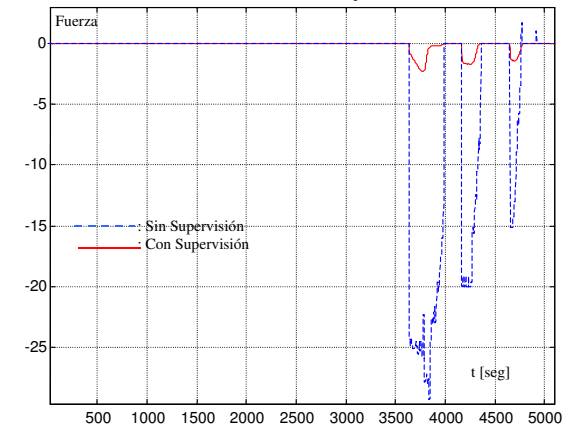


Fig. 14. Acción del Supervisor en la dirección  $x$ .

## 6. CONCLUSIÓN

En este artículo se presentó un controlador supervisor diseñado para un sistema de teleoperación robótico utilizando conceptos de sistemas híbridos. Este diseño está centrado principalmente en la interface que comunica el sistema continuo con el sistema de eventos discretos como así también en el controlador de eventos discretos, el cual fue obtenido utilizando como herramienta de diseño la teoría de autómatas.

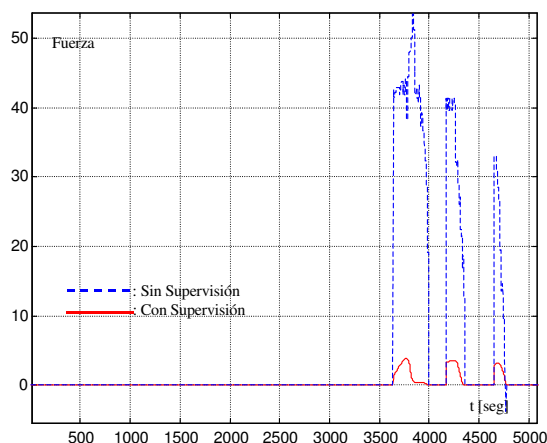


Fig. 15. Acción del Supervisor en la dirección y

Para tal fin se han definido las funciones y funcionales que son utilizados para la generación de los eventos. El diseño del autómatas está estrechamente vinculado con la tarea a desempeñar en el sistema de teleoperación y con el medio en el cual se desarrolla la misma. Esto implica tener cierto grado de conocimiento de la tarea y/o del medio. Es por ello que se considera conveniente tener a priori una trayectoria preprogramada y que el medio sea lo más estructurado posible. Esta estructura de control propuesta le confiere a la estación remota de trabajo un grado mayor de autonomía que se hace notorio cuando se produce una falla en la comunicación del sistema. Los resultados de simulación muestran un aceptable desempeño del sistema de teleoperación supervisado. Además es claro que este controlador protege al sistema remoto ante posibles comandos erróneos del operador humano cuando se realiza una tarea.

Como trabajos futuros se prevé la implementación de cámaras de video ubicadas convenientemente de forma tal de poder trabajar en ambientes no estructurados y adaptar el autómatas en línea. Se pretende además, repetir la experiencia considerando retardos de comunicación.

#### AGRADECIMIENTOS

Los autores quieren agradecer al Ministerio de Ciencia e Innovación (DPI 2009- 08778) y a la Comunidad de Madrid (RoboCity 2030 II). En memoria de Pepe.

#### REFERENCIAS

- Antsaklis P., Lemmon M., Stiver J., (1993). Hybrid System Modeling and Event Identification, Technical Report of the ISIS Group at the University of Notre Dame. ISIS – 93 – 002.
- Caltenco-Arciniega H.A., A. Melendez-Calderon, D.B. Popovic, J.E.Chong-Quero (2006). Reliability of Discrete Event Control at Coordination Level for a Powered Transfemoral Prosthesis. *16th IEEE Int. Conf. on Electronics, Communications and Computers*, p.6.
- Castro A. (1999). Modelado y Control telerobótico basado en modelo con información visual estéreo del robot SCARA. Reporte Técnico. Universidad Nacional de San Juan. Argentina. Instituto de Automática.
- Çavuşoglu M.C. (1997), *Control of a Telesurgical Workstation*. Thesis of Master of Science in Engineering. University of California at Berkeley.
- Chen, C.Y.P. and Wonham, W.M., (2001). Stable Supervisory Control of Flexible Manufacturing Systems with Fixed Supply and Demand Rates. *Int. J. of Production Research: Special Issue on Modeling, Specification and Analysis of Manufacturing Systems*. **39** (2), 347-368
- García C. Carelli R. Postigo J. Morales B. (2000a). Time Delay Compensation Structure for a Robotic Teleoperation System. *Proc. of the 4th IFAC International Symposium on Intelligent Components and Instruments for Control Applications*.
- García C., Morales B., Carelli R., Postigo J., Cury J. Supervisory Control for a Teleoperation System. Hybrid Control Approach (2000). Published CD of MCPL 2000. Grenoble. France.
- Hogan N. (1985). Impedance Control: An approach to manipulation: Part i, part ii, part iii. *ASME J. Dynamic Syst., Measurement, Control*, **107**..
- Jinbo Fu, Murat Yasar, Asok Ray (2004). Optimal Discrete Event Supervisory Control of Aircraft Gas Turbine Engines, *Proceedings of the 2004 American Control Conference*, Massachusetts. 5710-5715.
- Lin F., Ying H., Luan X., MacArthur R.D., Cohn J.A., Barth-Jones D., Crane L.R. (2004). Control of Fuzzy Discrete Event Systems and its Applications to Clinical Treatment Planning. *Proc. of the 43rd IEEE Conference on Decision and Control*, 519-524.
- Seidl M. (2006). Systematic Controller Design to Drive High-Load Call Centres. *IEEE Transactions on Control Systems Technology*, **14**(2), 216-223.
- Padala P, K. G. Shin, X. Zhu, M. Uysal, Z. Wang, S. Singhal A. Merchant, and K. Salem (2007). Adaptive control of virtualized resources in utility computing environments. *Proc. ACM SIGOPS European Conf. (EuroSys)*, pages 289–302.
- Postigo J., Mut V., Carelli R., (1994). Control Robusto Híbrido de Movimiento Fuerza para Robots incluyendo incertidumbre en el Medio. *VI Congreso Latinoamericano de Control Automático*.
- Prosser J., Selinsky J., Kwatny H. and Kam M. (1994), Supervisory Control of Electric Power Transmission Networks. *IEEE Transactions on Control Systems Technology*, 1104 –1110.
- Sheridan, T.B. (1992). *Telerobotics, Automation and Human Supervisory Control*, MIT Press USA.
- Stiver J. and Antsaklis P. (1992). Modeling and Analysis of Hybrid Control Systems. *Proc. of the 31st Conference on Decision and Control*. 3748-3751.
- Sukhan L. (1993). Modelling, Design and Evaluation of Advance Teleoperator Control Systems with Short Time Delay. *IEEE Transaction on Robotics and Automation*, **9**(5).
- Sutton R. (1990). *Modelling Human Operators in Control System Design*. J. Wiley & Sons Inc. New York.
- Wonham, W.M. (2003). Supervisory Control Theory: Models and Methods. *Proc. ATPN - Workshop on Discrete Event Systems Control, 24th International Conference on Application Theory of Petri Nets*. Netherlands, pp.1-14.