



Universitat de Girona

Trabajo de Investigación

Representación de la Dinámica del Cuerpo Físico de Agentes Físicos en Capacidades Atómicas

Autora:

Bianca M. Innocenti Badano

Tutor:

Dr. Josep Lluís de la Rosa i Esteva

Programa de Doctorado:

Informàtica Industrial / Tecnologies Avançades de Producció

Julio de 2000.

1	INTRODUCCIÓN.....	1
2	DEFINICIÓN DEL PROBLEMA.....	3
3	ESTADO DEL ARTE.....	8
3.1	AGENTES AUTÓNOMOS.....	8
3.1.1	<i>Introducción.....</i>	8
3.1.2	<i>Arquitecturas Agentes.....</i>	9
3.1.2.1	Agentes Basados en la Lógica o Agentes Deliberativos.....	10
3.1.2.2	Arquitecturas Reactivas.....	11
3.1.2.3	Arquitecturas BDI (Belief – Desire – Intention).....	12
3.1.2.4	Arquitecturas en Capas.....	14
3.1.3	<i>Lenguajes Agentes.....</i>	16
3.1.3.1	Programación Orientada a Agentes.....	16
3.1.3.1.1	Componentes del Estado Mental.....	17
3.1.3.1.2	Lenguaje para Belief, Obligation y Capability.....	17
3.1.3.1.3	Propiedades de los Componentes.....	18
3.1.3.1.4	Intérprete genérico.....	20
3.1.3.1.5	Funcionamiento de AGENTO.....	21
3.1.3.2	METATEM Concurrente.....	22
3.2	AGENTES FÍSICOS.....	22
3.3	ARQUITECTURA DPAA.....	25
3.4	RESUMEN.....	27
4	OBJETIVOS.....	32
5	CAPACIDADES ATÓMICAS.....	33
5.1	TEORÍA DE CONTROL.....	33
5.1.1	<i>Estabilidad Relativa.....</i>	34
5.1.2	<i>Estabilidad absoluta.....</i>	34
5.1.3	<i>Robustez.....</i>	34
5.1.4	<i>Sensibilidad.....</i>	35
5.1.5	<i>Errores de estado estable.....</i>	35
5.1.6	<i>Precisión.....</i>	37
5.1.7	<i>Rechazo de perturbaciones.....</i>	37
5.1.8	<i>Respuesta transitoria.....</i>	38
5.1.9	<i>Margen de fase y margen de ganancia.....</i>	39
5.1.10	<i>Índices de desempeño de controladores.....</i>	39
5.2	DEFINICIÓN DE LOS ATRIBUTOS.....	41
5.2.1	<i>Precisión.....</i>	44
5.2.2	<i>Sobrepico.....</i>	46
5.2.3	<i>Rapidez.....</i>	47

5.2.4	<i>Persistencia</i>	48
5.2.5	<i>Robustez</i>	50
5.2.6	<i>Agresividad</i>	52
5.2.7	<i>Esfuerzo de control</i>	52
5.2.8	<i>Coherencia</i>	53
5.2.9	<i>Identificador del sistema</i>	54
6	EJEMPLO	55
6.1	GENERACIÓN DE CAPACIDADES ATÓMICAS	55
6.1.1	<i>Planta</i>	56
6.1.2	<i>Controlador 1</i>	57
6.1.3	<i>Controlador 2</i>	61
6.1.4	<i>Controlador 3</i>	65
6.2	UTILIZACIÓN DE LAS CAPACIDADES EN LA DECISIÓN	70
6.3	PASE EN EL FÚTBOL	71
7	CONCLUSIONES	74
8	PROPUESTA DE TESIS	77
9	BIBLIOGRAFÍA	79
10	BIBLIOGRAFÍA COMPLEMENTARIA	81

1 Introducción.

La Inteligencia Artificial nació como un campo que pretendía replicar el nivel humano de inteligencia en una máquina. La IA tradicional se ha dedicado, principalmente, a tratar aspectos que requieran poca o nula intervención con el entorno y a conseguir una representación explícita y completa de su estado. En la manipulación simbólica clásica, la inteligencia se identifica con el razonamiento y éste, con el tratamiento basado en reglas de las estructuras simbólicas. Este tipo de inteligencia puede aplicarse para razonar sobre el mundo real, pero cómo se percibe el mundo o cómo ejercer interacciones con él son aspectos secundarios que no se tienen en cuenta en el proceso.

Al aplicar las técnicas desarrolladas por la IA a robots móviles, se descubrió que en ambientes muy estructurados y especialmente preparados se obtenían resultados poco satisfactorios, sobre todo los relacionados con el tiempo de ejecución de una determinada tarea. Así surgieron los requerimientos, según Brooks [2], que la inteligencia fuese reactiva respecto de aspectos dinámicos del entorno, que un robot móvil operase en escalas de tiempo similares a las humanas y la de los animales, y que la inteligencia fuese capaz de generar un comportamiento robusto frente a sensores imprecisos, un entorno impredecible y un mundo cambiante. Esto trajo como consecuencia nuevas tendencias en la IA, que insisten en que la inteligencia no puede separarse del cuerpo físico y a algunos investigadores como Brooks [2] y Mackworth [13], a considerar que la inteligencia se determina por la dinámica de interacción con el mundo.

Actualmente, se intentan desarrollar robots inteligentes combinando las técnicas de la IA con las teorías tradicionales de control, pero, aún no se ha podido definir una interfaz entre estas áreas y tampoco coordinar ambos niveles, ni analizar el comportamiento del sistema completo.

En la pasada década, esta nueva idea ha producido nuevos avances en la IA. Estos conceptos, que originariamente se aplicaron a la robótica, pero que actualmente se han extendido a otras áreas, se llaman generalmente sistemas multi-agentes. Tomando como referencia a Wooldrige y Jennings [11], el término agente se usa para definir un sistema software o hardware que cumple propiedades como la autonomía, la habilidad social, la reactividad, la proactividad, la

movilidad, la veracidad, la benevolencia y la racionalidad. Existen diferentes modelos de agentes, diversas arquitecturas y, asociados a ellas, varios lenguajes agente.

Entre estos últimos, se puede destacar el lenguaje AGENT0 de Shoham [10], que utiliza nociones *mentales*, como las creencias, las capacidades y las decisiones, para caracterizar al agente. Las capacidades determinan las cosas que el agente es capaz de, o puede hacer, por lo que las necesita en el momento de tomar una decisión, fruto de compromisos.

Cuando el agente posee un cuerpo físico, la decisión de qué acción realizar se ve afectada por la dinámica del cuerpo y en algunos casos ésta, aunque fuera heurísticamente posible, puede conducir a consecuencias no deseadas.

El motivo de este trabajo de investigación es encontrar la manera de utilizar las capacidades del lenguaje AGENT0 para representar el conocimiento dinámico que se tiene sobre el cuerpo físico del agente, y posteriormente usarlo para tomar decisiones físicamente realizables.

2 Definición del Problema.

En el lenguaje AGENT0, un agente es una entidad cuyo estado consiste en un conjunto de *capacidades* (*capacities* - cosas que el agente puede hacer), en un conjunto inicial de *creencias* (*beliefs*), en un conjunto inicial de *compromisos* (*commitments*) y en un conjunto de *reglas de compromisos* (*commitment rules*).

Los componentes claves, que determinan como actúa un agente, son las reglas de compromisos. Cada una de ellas, contiene una *condición de mensaje*, una *condición mental* y una acción. Para comprobar si tales reglas se disparan, la condición de mensaje se compara con los mensajes que ha recibido el agente y la condición mental, con las creencias del agente, de esta manera se sintetizan comportamientos reactivos - proactivos . Si la regla se dispara, el agente queda comprometido con la acción (adquiere un *commitment*).

El futuro del mundo del agente, queda establecido, en cualquier instante de tiempo, por el pasado y por las acciones realizadas por todos los agentes que en él están. Las acciones se originan en las *decisiones* o *elecciones* del agente. Es decir, que algunos hechos son verdaderos por razones naturales, y otros, porque el agente así lo decide. Las *decisiones* están restringidas, aunque no fijadas, por las *creencias* del agente respecto al estado del entorno (pasado, presente o futuro), por el estado mental de los otros agentes, por las *decisiones* previas y por las *capacidades* que tienen él y los demás agentes.

Un ejemplo de una regla de compromiso en AGENT0 sería:

```
COMMIT(
    (agente, REQUEST, DO (tiempo,acción) ),  →  condición de mensaje
    (B,
        [ahora, Friend agente] AND
        CAN (self, acción) AND
        NOT [tiempo, CMT (self, ninguna_acción)]
    ),  →  condición mental
    self,
    DO (tiempo,acción)
)
```

Esta regla se lee como:

*Si recibo un mensaje de **agente** que me pide (**request**) que haga (**do**) una **acción** en un **tiempo** determinado, y creo (**believe B**) que:*

- ***agente** es actualmente un **amigo**;*
- *Puedo (**can**) hacer (**do**) la **acción**;*
- *Y en el **tiempo** determinado, no estoy comprometido (**cmt**) para realizar otra acción, Entonces me **comprometo** para **hacer** la **acción** en el **tiempo** requerido.*

Con el propósito de cumplir con sus compromisos, el agente debe estar seguro que éstos son realizables, por lo que antes de comprometerse a ejecutar una acción, contrasta lo que se le pide hacer con sus creencias y sus capacidades, como se observa en el ejemplo anterior.

En agentes físicos (tienen un cuerpo físico, por ejemplo, un robot) la realización de una acción depende también de la dinámica de este cuerpo físico.

Para saber qué puede o no hacer, un agente físico debe tener alguna clase de conocimiento físico, lo que significa que las entradas y salidas físicas de y al entorno, deben estar representadas en la base de conocimiento (beliefs) de cada agente.

Las capacidades parecen el “estado mental” adecuado para introducir el conocimiento físico que se tiene a priori de la dinámica del cuerpo del agente. En el momento de tomar una decisión, el agente verificará si su cuerpo puede realizar la acción propuesta y así producir las modificaciones esperadas en el entorno.

De la Rosa et al [6], presentan un ejemplo de cómo influye la dinámica en la decisión del agente. El ejemplo trata de un convoy de dos vehículos cuyas dinámicas son diferentes. Los vehículos están separados una determinada distancia y van a una velocidad establecida. La idea es mantener constante una distancia de separación entre ellos, aún cuando el vehículo guía (adelante) decida cambiar su velocidad de cruce. La responsabilidad de mantener la distancia constante, recae en el vehículo de atrás, que cambia su dinámica de acuerdo con ésta.

El problema se presenta cuando la consigna de distancia de separación de los vehículos no es lo suficientemente grande, y ante un cambio de velocidad del guía, se producen colisiones. La respuesta tradicional a este problema ha sido la de aumentar la distancia de separación hasta el punto en el cual las colisiones se eliminan.

Una posible propuesta en el campo de control, sería la de utilizar un tipo de controlador (feedforward) que compense los cambios de consigna que se sabe que existirán. Existen varias maneras de implementar el control feedforward. Una sería no tener en cuenta la dinámica del vehículo guía, pero de alguna manera conocer exactamente la velocidad que tiene en cada instante de tiempo, y así compensar los cambios que en ella se produzcan. Esto exigiría una comunicación entre los vehículos o un sensor que midiese la velocidad del vehículo guía. Otra solución sería conocer la dinámica del vehículo guía y compensar directamente la salida. Este caso no exige comunicación ni medición de la velocidad del coche de delante pero tiene un inconveniente tendría que haber un controlador feedforward para cada dinámica externa diferente. Directa o indirectamente, se tiene que incluir conocimiento sobre la dinámica del vehículo guía.

Por otra parte, la solución que se propone en el campo de los agentes, es la de emplear agentes deliberativos, para que antes de producirse el cambio de consigna, el vehículo guía informe a los demás de la acción que ejecutará. Pero, como se explicará a continuación, no se tiene en cuenta la dinámica de los vehículos, por lo que, aún cuando todos los vehículos llegan a un acuerdo respecto del cambio de velocidad, se producen colisiones si la distancia de separación no es lo suficientemente grande.

En la Figura 2-1, presentada en [6], se esquematiza la distancia de separación entre los vehículos que forman el convoy cuando el agente guía decide frenar. En este caso, los agentes son reactivos, es decir, el agente que sigue al guía actúa de acuerdo con los cambios que percibe en su entorno. En ella se puede observar como la distancia de separación disminuye hasta la colisión y que, cuando la respuesta de los vehículos alcanza su estado permanente, existe una separación entre ellos (aunque no es la especificada). En este ejemplo, se muestra claramente que el transitorio de la respuesta de los vehículos, es decir, su dinámica, modifica completamente en el resultado esperado de la acción; lo que induce a pensar que es necesario que el agente tenga en cuenta su respuesta dinámica en el momento de la decisión.

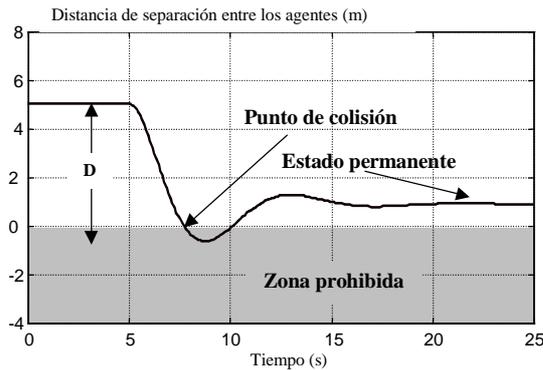


Figura 2-1: Distancia de separación entre los agentes.

La solución propuesta en [6] para este problema es la de incluir en las capacidades del agente el conocimiento sobre la dinámica de su cuerpo físico y utilizarlo al momento de la decisión.

Desgraciadamente, [6] compara como afecta a la decisión, que los agentes sean deliberativos y reactivos; pero no la diferencia que pueda existir al realizar una acción, entre dos convoys de vehículos con agentes deliberativos, uno de los cuales tiene en cuenta la dinámica y el otro no.

En [6] se ha hecho una primera aproximación de cómo deberían completarse las capacidades y de cómo buscar la que mejor se adecua a las especificaciones. Los resultados no han sido del todo satisfactorios pero han demostrado la complejidad que existe tanto en la elección de los atributos que forman las capacidades como en su posterior tratamiento por el agente.

El problema más evidente de la aproximación que se propone, es el de decidir qué parámetros de la respuesta del cuerpo físico del agente son necesarios para representar su dinámica. Estos parámetros deben tener, entre otras, las siguientes características: ser computacionalmente entendibles (procesables) por el agente y ser comparables (para poder contrastar las capacidades al momento de decidir).

El propósito del presente trabajo, consiste en encontrar la manera adecuada de representar la dinámica del cuerpo físico de un agente en términos entendibles por éste. Para ello, es necesario determinar, en el ámbito de control, cuáles son los parámetros que mejor describen el comportamiento de un sistema controlado. Éstos dependerán de la clase de sistema que se

desea controlar, el tipo de control empleado y de las especificaciones de control que se esperan cumplir. Por otra parte, hay que adecuarlos para que sean tratables computacionalmente por el agente.

Debido a la complejidad del problema, en el presente trabajo se restringirá el estudio a sistemas SISO (de una entrada una salida), previéndose la ampliación a sistemas MIMO (múltiple entrada – múltiple salida), como trabajo futuro.

3 Estado del Arte.

3.1 Agentes Autónomos.

3.1.1 Introducción.

Un *agente*, según Wooldridge [12], es un sistema computacional que está *situado* en algún *entorno*, y que es capaz de realizar *acciones autónomas* en su entorno con el propósito de conseguir los objetivos para los que fue diseñado. Realizar *acciones autónomas* significa que los agentes operan sin la intervención directa de humanos u otros sistemas, y tienen el control sobre sus estados internos y sobre su comportamiento.

Normalmente un agente posee un repertorio de acciones que representan la capacidad de efecto (effectoric capacity), es decir, su habilidad para modificar su entorno. Como no todas las acciones pueden ejecutarse siempre, existen unas precondiciones (pre-conditions) asociadas a ellas, que definen las posibles situaciones en las cuales pueden aplicarse.

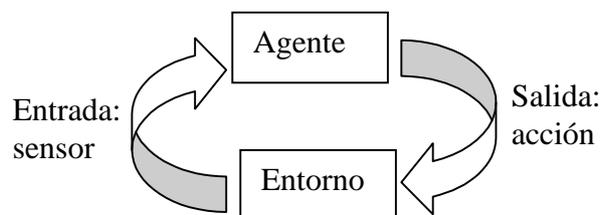


Figura 3-1: El agente y su entorno.

La Figura 3-1 muestra una vista abstracta a alto nivel de un agente. Un agente típicamente percibirá su entorno (con sensores físicos en el caso que el agente esté situado en una parte del mundo real, o con sensores software, si son agentes software) y dispondrá de un repertorio de acciones que puede ejecutar para modificar su entorno, el cual parecerá responder de forma no determinística a la ejecución de estas acciones. Un entorno no determinístico es aquel en el cual, desde el punto de vista del agente, una misma acción ejecutada dos veces en condiciones aparentemente idénticas, produce efectos completamente diferentes.

Un agente inteligente es aquel capaz de ejecutar acciones autónomas *flexibles* para cumplir con sus objetivos, y por flexible se entiende que posea las tres características siguientes:

- **Reactividad:** los agentes inteligentes son capaces de percibir su entorno y responder de manera oportuna a los cambios que en él ocurren.
- **Proactividad:** los agentes inteligentes no solamente actúan en respuesta a su entorno, sino que exhiben un comportamiento intencionado tomando *la iniciativa*.
- **Habilidad Social:** los agentes inteligentes son capaces de interactuar con otros agentes (y con seres humanos) a través de alguna clase de lenguaje de comunicación agente.

Desde un punto de vista más estricto, un agente también debe tener los siguientes atributos:

- **Mobilidad:** es la habilidad que tienen los agentes de moverse a través de su entorno.
- **Veracidad:** se asume que un agente no comunicará información que sabe es falsa.
- **Benevolencia:** es la suposición que el agente no tendrá objetivos en conflicto y que cada agente tratará de hacer todo lo que se le pida.
- **Racionalidad:** es la presunción que el agente actuará para cumplir con sus objetivos y no de manera contraria, mientras que sus creencias así se lo permitan.

3.1.2 Arquitecturas Agentes.

El problema principal que enfrenta un agente es el de decidir qué acción debería realizar de modo de satisfacer convenientemente sus metas. Las *arquitecturas agentes* son, en realidad, arquitecturas software para sistemas de toma de decisiones que están emplazadas en un entorno. Principalmente existen cuatro clases de arquitecturas, que son:

- **Agentes basados en la lógica:** en los cuales el proceso de decisión se realiza a través de la deducción lógica.

- **Agentes reactivos:** la decisión se implementa de modo que existe una correlación directa entre la situación y la acción.
- **Agentes BDI (Belief – Desire – Intention):** en los que la decisión depende de la manipulación de estructuras de datos que representan las creencias, los deseos y las intenciones del agente.
- **Arquitecturas en capas:** en las que la decisión se realiza a través de varias capas software, cada una de las cuales tiene más o menos razonamiento explícito acerca del entorno, en diferentes niveles de abstracción.

3.1.2.1 Agentes Basados en la Lógica o Agentes Deliberativos.

La aproximación tradicional para construir sistemas de inteligencia artificial (AI simbólica) sugiere que un comportamiento inteligente puede generarse dándole a un sistema una representación simbólica del entorno y el comportamiento deseado, y manipulando sintácticamente esta representación. La representación simbólica del entorno se consigue mediante fórmulas lógicas, y la manipulación sintáctica, a través de una deducción lógica o demostración de teoremas.

En tales agentes, se asume que el estado es una base de datos de fórmulas de predicados lógicos de primer orden. La base de datos es la información que posee el agente sobre el entorno. El proceso de toma de decisión del agente se modela a través de un conjunto de reglas de deducción (reglas de inferencia). Así, el comportamiento del agente queda determinado por las reglas de deducción y la base de datos actual.

Las aproximaciones basadas en la lógica son elegantes y tienen una semántica clara (lógica). Pero, tienen varias desventajas:

- **No son instantáneos:** La complejidad computacional inherente a la demostración de teoremas hace cuestionable si los agentes deliberativos pueden operar eficazmente en entornos con restricciones temporales. La toma de decisiones en este tipo de agentes se

basa en la suposición de un entorno que no cambia significativamente mientras el agente está decidiendo que hacer, y en que la acción, que era racional en el momento que comenzó el proceso de decisión, sea racional cuando éste concluya.

- **Representación del entorno:** Este tipo de agentes, utiliza una representación simbólica del entorno que típicamente, es un conjunto de fórmulas en el lenguaje de representación agente. Para entornos complejos, dinámicos y posiblemente físicos, este tipo de correlación no es obvia.
- **Razonamiento sobre el entorno:** Hasta la representación del conocimiento procedimental simple puede ser no intuitivo y laborioso en la lógica tradicional. El razonamiento sobre la información temporal (como cambia una situación en el tiempo), resulta ser extremadamente difícil. Éste es el caso de entornos complejos, dinámicos y físicos.

3.1.2.2 Arquitecturas Reactivas.

Otra arquitectura agente es la denominada del comportamiento (desarrolla y combina comportamientos individuales), situada (los agentes están realmente situados en un entorno) y reactiva (porque los sistemas reaccionan al entorno sin razonar acerca de él).

La toma de decisión del agente se realiza a través de un conjunto de comportamientos que cumplen tareas; cada comportamiento puede pensarse como una función, que toma continuamente las entradas de percepción y las correlaciona con acciones a realizar. Cada uno de estos módulos de comportamientos se diseñan para conseguir alguna tarea en particular. Un punto importante a destacar es que, estos módulos no incluyen representaciones simbólicas complejas ni hacen ningún tipo de razonamiento simbólico.

Otra característica de este tipo de agentes, es que varios comportamientos pueden dispararse al mismo tiempo. Obviamente debe existir un mecanismo para elegir entre las diferentes acciones seleccionadas. Para solucionar este problema, Brooks propone arreglar los módulos en una jerarquía *subsumption* (*categorización*), con los comportamientos dispuestos en capas. Las capas inferiores en la jerarquía pueden inhibir a las capas superiores; cuanto más bajo nivel tenga la capa, más prioritaria es.

Existen ventajas evidentes en las aproximaciones reactivas: simplicidad, economía, flexibilidad computacional, robustez contra fallas, y elegancia; mas, existen algunos problemas no resueltos:

- Si los agentes no utilizan modelos de su entorno, deben tener suficiente información disponible de su entorno local, para poder determinar una acción aceptable.
- Ya que los agentes puramente reactivos toman decisiones basadas en información local, es difícil de ver como puede, este tipo de toma de decisiones, tener en cuenta información no local.
- Es difícil entender como agentes reactivos puros pueden diseñarse para aprender por medio de la experiencia y a mejorar su desempeño a lo largo del tiempo.
- No existe una metodología para construir agentes cuyo comportamiento global emerja de la interacción de los comportamientos integrantes, cuando se emplaza al agente en su entorno. Se debe utilizar un proceso laborioso de experimentación, prueba y error para diseñar al agente.
- Mientras que se pueden generar agentes efectivos con un bajo número de comportamientos (típicamente menor de 10), es muy difícil construirlos con muchas capas. La dinámica de las interacciones entre los diferentes comportamientos se hace muy compleja de entender.

3.1.2.3 Arquitecturas BDI (Belief – Desire – Intention).

Los componentes básicos de esta arquitectura son las creencias, los deseos y las intenciones del agente; las funciones que representan su deliberación; y el razonamiento de fines y medios.

Este tipo de arquitectura tiene sus raíces en la tradición filosófica del entendimiento del razonamiento práctico, que es el proceso de decidir, momento a momento, qué acción ejecutar para cumplir con los objetivos fijados.

El razonamiento práctico involucra dos procesos importantes: decidir qué metas se desean conseguir, proceso que se conoce como deliberación; y cómo se lograrán estas metas, procedimiento que se denomina razonamiento de fines y medios (means-ends reasoning).

El proceso de decisión empieza típicamente tratando de comprender que *opciones* están disponibles; una vez generado este conjunto de alternativas, se debe elegir entre ellas y *comprometerse* (*commit*) con una; esta opción escogida se convierte en una *intención*, la cual determina las acciones del agente. Las intenciones focalizan el razonamiento práctico futuro del agente; cuando se tiene una intención en particular, se descartan todas aquellas opciones que sean inconsistentes con la intención. Además, una vez adoptada una intención, el agente debe *perseverar* (*persist*) en ésta, sólo debe rectificarla cuando la razón por la cual tenía la intención ha cambiado; o cuando el agente sabe con certeza que no podrá cumplir con ella.

Finalmente, las intenciones están estrechamente relacionadas con las creencias acerca del futuro. Cuando tiene una intención, el agente al menos debe creer que tiene una gran posibilidad de cumplir con ella.

Un problema clave en el diseño del razonamiento práctico de los agentes es el de adquirir un buen balance entre los diferentes intereses. A veces, los agentes deberían abandonar algunas de sus intenciones (ya sea porque cree que son inalcanzables, porque ya las ha conseguido o porque la razón por la que tenía la intención ya no está presente). Esto conlleva a que el agente deba reconsiderar sus intenciones cada tanto; lo que presenta un dilema que esencialmente es el de crear un balance entre el comportamiento pro-activo (intencionado) y el reactivo (condicionado a eventos):

- Un agente que no se detiene lo suficientemente seguido a reconsiderar sus intenciones, continuará pretendiendo cumplir con ellas, aún cuando esté claro que no las puede consumir o la razón por la cual las tiene ya no está presente.

- Un agente que está constantemente reconsiderando sus intenciones puede dedicar poco tiempo al trabajo necesario para conseguirlas y por lo tanto corre el riesgo de nunca cumplirlas.

La dinámica del entorno condiciona este balance. Cuanto más dinámico sea el entorno, la habilidad de reaccionar a cambios modificando las intenciones se hace más importante.

Las razones porque este modelo es atractivo son: primero porque es intuitivo – se tiene un conocimiento informal de los términos creencias, deseos e intenciones; el método de decisión se parece al razonamiento práctico usado diariamente; segundo da una descomposición funcional clara, la cual indica que clase de subsistemas se pueden requerir para construir el agente. La dificultad principal es, como siempre, la de saber como implementar eficientemente estas funciones.

3.1.2.4 Arquitecturas en Capas.

Dado el requerimiento que un agente sea capaz de tener comportamientos reactivos y pro-activos, una descomposición clara es crear subsistemas separados para tratar estos tipos de comportamientos diferentes. La idea lleva naturalmente a una clase de arquitectura en la cual varios subsistemas son jerarquizados en capas que interactúan entre sí.

Típicamente habrá dos capas para tratar con el comportamiento reactivo y pro-activo, respectivamente. Sin embargo, cuantas más capas haya, más útil es la topología de tales arquitecturas, por el flujo de información y control que hay entre ellas. En general, se pueden identificar dos tipos de flujos de control:

- **Capas horizontales:** Cada capa de software está directamente conectada a las entradas sensoriales y a las salidas actuadoras. En efecto, cada capa actúa por sí misma, como un agente, produciendo sugerencias de qué acción realizar (Figura 3-2 (a)).
- **Capas verticales:** Una capa se encarga de manipular las entradas sensoriales y las salidas actuadoras (Figura 3-2 (b) y (c)).

La ventaja de las arquitecturas en capas horizontales es su simplicidad conceptual: si se necesita que un agente presente n tipos diferentes de comportamientos, entonces hace falta implementar n capas diferentes. No obstante, debido a que las capas están cada una compitiendo con las otras para generar la acción sugerida, existe el peligro de que el comportamiento global del agente no sea coherente. Para asegurar que las arquitecturas en capas horizontales sean consistentes, generalmente incluyen una función de mediación, la cual decide que capa tiene el “control” del agente en un instante determinado. La necesidad de este control centralizado es problemática ya que obliga al diseñador a considerar todas las interacciones posibles entre las capas. Éste también, introduce un cuello de botella en el algoritmo de decisión del agente.

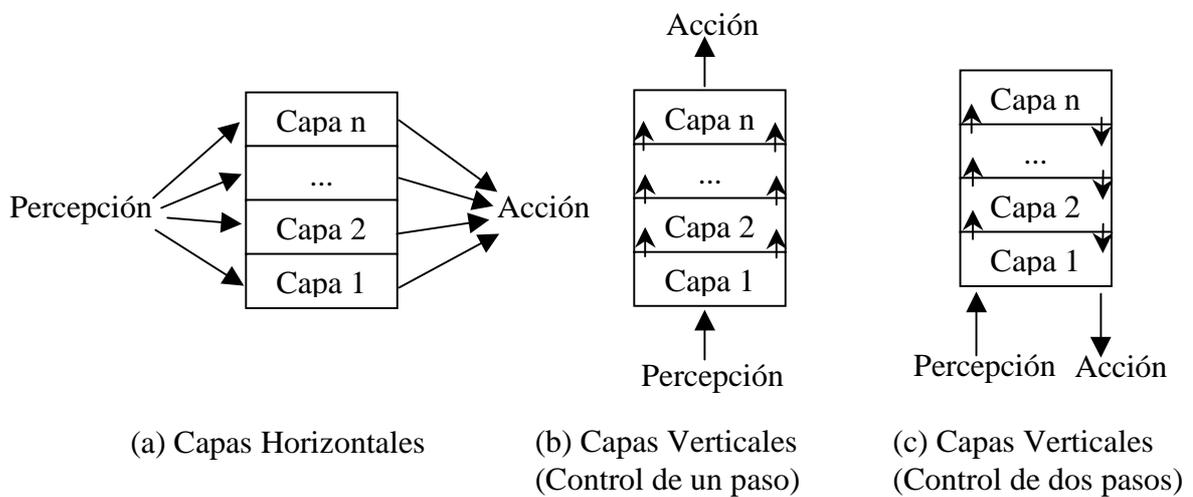


Figura 3-2: Flujo de control e información en tres arquitecturas agentes por capas.

Algunos de estos problemas se solucionan con la arquitectura en capas verticales. Éstas pueden dividirse en arquitecturas de capas verticales de un o dos pasos de control. En las primeras, el control fluye secuencialmente a través de cada capa, hasta que la final genera la acción de salida. En las segundas, la información fluye hacia los niveles superiores de la arquitectura mientras que el control fluye hacia las capas inferiores. En ambas, la complejidad de las interacciones entre las capas se reduce considerablemente. Sin embargo, esta simplicidad es a cambio de flexibilidad: para poder lograr que la arquitectura en capas verticales tome una decisión, el control debe pasar por cada una de las diferentes capas. Esto no es tolerante a fallos, y una falla en cualquiera de las capas, trae graves consecuencias en el desempeño del agente.

3.1.3 Lenguajes Agentes.

A medida que la tecnología agente se establece, se espera disponer de una serie de herramientas para el diseño y construcción de los sistemas basados en agentes. A continuación se resumirán los dos lenguajes agentes más conocidos, haciendo hincapié en el lenguaje Agent0 de Shoham [10].

3.1.3.1 Programación Orientada a Agentes.

Por lenguaje agente, se entiende a un sistema que permite programar sistemas de computadoras hardware o software en términos de algunos conceptos desarrollados por los teóricos de los agentes. Intuitivamente puede verse como una especialización de la programación orientada a objetos (P.O.O.). Mientras que, por un lado, la P.O.O. propone concebir aplicaciones software compuestas por módulos o estructuras (objects) capaces de intercambiar información entre ellos y que tienen maneras individuales para manejar los mensajes entrantes por medio de los métodos, la programación orientada a agentes (A.O.P.) especializa la P.O.O. extendiendo el estado de los objetos, considerados ahora como agentes, con estados mentales (que consisten en componentes tales como creencias, capacidades y decisiones). Además, la A.O.P. contiene métodos más avanzados y especializado de paso de mensajes entre agentes, como por ejemplo: informar, pedir, ofrecer, aceptar, rechazar, competir y asistirse entre ellos.

La idea principal de la A.O.P. es hacer la programación directa de los agentes en términos de las nociones intencionales y mentalistas que los teóricos de agentes han desarrollado para representar las propiedades de los agentes.

Se propone, que un sistema A.O.P. completamente desarrollado tenga tres componentes:

- Un sistema lógico para definir el estado mental del agente.
- Un lenguaje de programación interpretado para programar agentes.
- Un proceso de “agentificación” para compilar programas agentes en sistemas ejecutables de bajo nivel.

Un primer intento de A.O.P. es el lenguaje AGENT0 de Shoham. Para Shoham, un agente es una entidad cuyos estados se ven constituidos por componentes mentales como *beliefs*, *capabilities*, *choices* y *commitments*.

3.1.3.1.1 Componentes del Estado Mental.

En cualquier punto en el tiempo, el futuro se determina por dos factores: la historia pasada, y las acciones actuales del agente. Las acciones de un agente se determinan por sus decisiones o elecciones. En otras palabras, algunos hechos son verdaderos por razones naturales y otros, porque el agente decide hacerlos. Las decisiones están restringidas lógicamente, aunque no determinadas, por las creencias del agente; estas creencias se refieren al estado del mundo (en el pasado, presente o futuro), al estado mental de otros agentes, y a las capacidades del agente y a las de los otros. Por ejemplo, si un robot cree que es incapaz de pasar a través de una puerta estrecha, no decidirá atravesarla. Las decisiones están restringidas también, por las decisiones previas, es decir, por ejemplo, el robot no puede decidir estar en la habitación número cinco en cinco minutos, si ya ha decidido estar en la tercera en ese momento.

Esta perspectiva motiva la introducción de dos categorías mentales básicas, creencia (*belief*) y decisión (*decision*), y una tercera categoría las capacidades (*capabilities*), que no son un constructor mental *per se*.

3.1.3.1.2 Lenguaje para Belief, Obligation y Capability.

- **Time:** Básico para las categorías mentales; se creen cosas acerca del tiempos diferentes y en tiempos diferentes.
Ejemplo: sostener(robot,copa)^t: el robot sostiene la copa en el tiempo t.
- **Acción:** Se llevan a cabo en diferentes puntos en el tiempo y dependiendo de las circunstancias en el tiempo en que se hacen, tienen ciertos efectos. Dado que las acciones son hechos, son, también, instantáneas. Esto es una limitación en este lenguaje.

- **Creencia:** Se aumenta el lenguaje con el operador modal B (belief). Como se dijo, el agente cree cosas en tiempos diferentes y acerca de tiempos diferentes.

$B_a^t \varphi$: donde a es un agente, t el tiempo, y φ una sentencia.

Ejemplo:

$B_a^3 B_b^{10} \text{like}(a,b)^7$: en el tiempo 3, el agente **a** cree que en el tiempo 10 el agente **b** creerá que en el tiempo 7 a **a** le gusta **b**.

- **Obligación:** Se introduce un nuevo operador modal OBL; $OBL_{a,b}^t \varphi$ significa que en el tiempo t, **a** está obligado o comprometido con **b** acerca de φ .
- **Decisión:** La libertad de elegir entre varias acciones posibles es un punto clave de la noción de “agencia”. La actual definición de obligación provee de una alternativa, sin embargo, la decisión se define como una simple obligación o compromiso consigo mismo.

$DEC_a^t \varphi =_{\text{def}} OBL_{a,a}^t \varphi$

- **Capacidad:** Intimamente relacionada a la noción de agencia, se encuentra la de capacidad. Por ejemplo, se puede decidir mover un brazo, pero si no se es capaz, entonces no se lo moverá.

$CAN_a^t \varphi$: en el tiempo t el agente **a** es capaz de φ .

Ejemplo:

$CAN_{\text{robot}}^5 \text{open}(\text{door})^8$: en el tiempo 5 el robot asegura que la puerta está abierta en el tiempo 8; pero en el tiempo 6 podría no tener ya esa capacidad.

Se puede definir ABLE como la versión “inmediata” de CAN.

$ABLE_a \varphi =_{\text{def}} CAN_a^{\text{tiempo}(\varphi)} \varphi$; donde $\text{tiempo}(\varphi)$ es el mayor tiempo que ocurre φ .

3.1.3.1.3 Propiedades de los Componentes.

Se harán las restricciones necesarias en las modalidades expuestas, para establecer una correspondencia entre la definición formal y el sentido común. Se especificarán las propiedades asumidas acerca de las modalidades.

- **Consistencia Interna:** Se asume que tanto las creencias como las obligaciones son consistentes internamente (no hay contradicciones entre ellas).
 - $\forall a, t: \{\varphi: B_a^t \varphi\}$ es consistente.
 - $\forall a, t: \{\varphi: OBL_{a,b}^t \varphi, \text{ para algún } b\}$ es consistente.

- **Buena Voluntad:** Se asumirá que los agentes se comprometerán solamente en lo que se crean capaces de hacer.
 - $\forall t, a, b, \varphi: OBL_{a,b}^t \varphi \supset B_a^t ((ABLE_a \varphi) \wedge \varphi)$

- **Introspección:** Aunque en general no se asume que los agentes tengan unas capacidades totalmente introspectivas, se asume que están conscientes de sus obligaciones:
 - $\forall t, a, b, \varphi: OBL_{a,b}^t \varphi \equiv B_a^t OBL_{a,b}^t \varphi.$
 - $\forall t, a, b, \varphi: \neg OBL_{a,b}^t \varphi \equiv B_a^t \neg OBL_{a,b}^t \varphi.$

Por otra parte, no se asume que los agentes sean necesariamente conscientes de los compromisos hechos con ellos.

- **Persistencia del Estado Mental:** Se han puesto restricciones sobre las actitudes mentales en un único instante de tiempo. Se discutirá como cambian o persisten en el tiempo los estados mentales.

Se colocará una fuerte restricción a las creencias: se asumirá que los agentes tienen una memoria perfecta de, y fe en, sus creencias; y solamente dejarán de creer si aprenden un hecho contradictorio. Las creencias persisten, entonces, por defecto. Aún más, se asumirá que la ausencia de creencias también persistirá por defecto, aunque con una sutil diferencia: si un agente no cree en un hecho a un tiempo determinado (opuesto a creer la negación del hecho), entonces la única razón que le llevará a creer en el hecho será que lo aprenda.

Las obligaciones deben ser también persistentes, sino no son obligaciones. Sin embargo, contraria al caso de las creencias, la persistencia no es absoluta. Aunque por defecto las obligaciones persisten, hay condiciones bajo las cuales se revocan. Estas condiciones

presumiblemente incluyen la liberación explícita de un agente por la parte con la que está obligado; o alternativamente que se de cuenta de que ya no es capaz de cumplir con la obligación.

Ya que la decisión se define en términos de la obligación, hereda la persistencia por defecto. Mientras que el agente no puede revocar unilateralmente las obligaciones que tiene hacia otros, puede cancelar las obligaciones que se tienen para con él, incluyendo las consigo mismo (decisiones). Un agente es entonces, libre de modificar una decisión existente, pero a menos que lo haga explícitamente, la decisión permanecerá.

Finalmente, las capacidades tienden a no fluctuar ampliamente. De hecho, se asume que las capacidades son fijas. Lo que un agente puede hacer en un tiempo también lo puede hacer en otro. Sin embargo se permitirá limitar una capacidad de acción bajo ciertas condiciones que se mantienen en el momento de la acción.

3.1.3.1.4 Intérprete genérico

El papel de un programa agente es controlar la evolución del estado mental del agente. Las acciones ocurren como un efecto secundario al hecho de asumir un compromiso (es decir, obligarse) con una acción cuyo tiempo ha llegado.

El comportamiento del agente, es en principio, bastante sencillo. Cada agente itera los dos siguientes pasos a intervalos regulares:

- 1) Lee el mensaje actual y actualiza su estado mental, incluyendo sus creencias y compromisos.
- 2) Ejecuta los compromisos para el tiempo actual que posiblemente resulten en un cambio en las creencias más adelante.

Las acciones con las cuales el agente se puede comprometer incluyen las comunicativas como informar y pedir; y las acciones privadas arbitrarias.

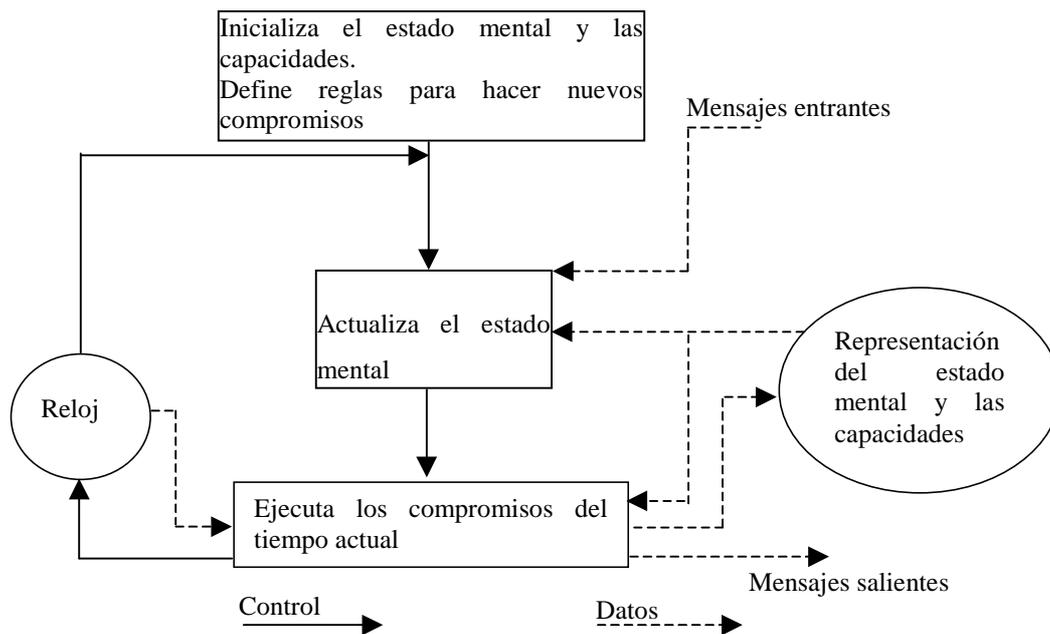


Figura 3-3: Diagrama de flujo de un intérprete agente genérico.

3.1.3.1.5 Funcionamiento de AGENT0.

Como se ha dicho un agente se especifica en términos de un conjunto de capacidades (cosas que puede hacer el agente), un conjunto de creencias y compromisos iniciales, y un conjunto de reglas de compromiso. El componente principal que determina como actúa el agente es el conjunto de reglas de compromiso. Cada regla de compromiso contiene una condición de mensaje, una condición mental y una acción. Para determinar si esta regla se dispara, la condición de mensaje se compara con otros mensajes que el agente ha recibido; la condición mental se compara con las creencias del agente. Si se dispara la regla, entonces el agente se compromete con la acción. Las acciones pueden ser privadas, correspondiendo a una subrutina ejecutada internamente o comunicativas, es decir, envío de mensajes. Los mensajes están restringidos a ser uno de tres tipos: “request” o “unrequest” para realizar o abstenerse de realizar acciones y los mensajes “inform” que se utilizan para pasar información. Los mensajes request y unrequest resultan en una modificación de los compromisos del agente mientras que los “inform”, en un cambio de las creencias de los agentes.

3.1.3.2 METATEM Concurrente.

Un sistema METATEM concurrente, (lenguaje desarrollado por Fisher) contiene un número de agentes que se ejecutan de manera concurrente, cada uno de los cuales puede comunicarse con sus pares vía un broadcast asincrónico de mensajes. Cada agente se programa dándole una especificación lógica temporal del comportamiento que debe tener. Una especificación del agente se ejecuta directamente para generar su comportamiento. La ejecución del programa del agente corresponde a construir iterativamente un modelo lógico temporal de la especificación del agente.

La semántica lógica del METATEM concurrente está estrechamente relacionada con la semántica de la lógica temporal misma. Esto significa que, entre otras cosas, la especificación y la verificación de los sistemas METATEM concurrente es una proposición realista.

Este lenguaje tiene reglas de la forma $P_i \Rightarrow F_i$, donde P_i es una fórmula lógica temporal que se refiere solamente al pasado o presente, y F_i es una fórmula lógica temporal que se refiere al presente o futuro. Las reglas se comparan con la historia interna guardada, y si se encuentra una equivalencia, la regla se dispara. Si una regla se dispara, se instancia cada variable en la parte de tiempo futuro (F_i), y entonces esta parte, se convierte en un compromiso que el agente intentará cumplir. Satisfacer un compromiso significa hacer algún predicado verdadero dentro del agente.

3.2 Agentes Físicos.

Los primeros trabajos de la IA se concentraron en juegos, problemas geométricos, álgebra simbólica, prueba de teoremas y otra clase de sistemas formales. En cada caso, la semántica de los dominios era bastante simple. Al final de los 60's y comienzo de los 70's, la técnica de representar al mundo con bloques se hizo popular entre los investigadores de IA. Tenía una semántica uniforme y simple. La clave de su éxito consistió en representar el estado del mundo completa y explícitamente. Las técnicas de búsqueda podían utilizarse para hacer planificaciones dentro de este mundo bien conocido. También se podía aprender. Eventualmente, la crítica sacó a relucir que los bloques eran un "mundo de juguetes" y que dentro de él, había simplemente soluciones con propósitos especiales a los que debían

considerarse como problemas más generales. Pronto apareció un nuevo eslogan: “La buena representación es clave para la IA”. La idea era que representando explícitamente sólo los hechos pertinentes, la semántica del mundo se reduciría a un simple sistema cerrado otra vez. La abstracción aplicada a los detalles importantes solamente, simplificaba los problemas.

La única entrada en la mayoría de los programas de IA estaba restringida a un conjunto de afirmaciones deducidas de la realidad por humanos. Los problemas de reconocimiento, comprensión espacial, tratamiento del ruido en los sensores, modelos parciales, etc. se ignoraban. Estos problemas se relegaban al dominio de entradas de cajas negras. Existen evidencias psicofísicas que sugieren que todos ellos están íntimamente ligados con la representación del mundo usada por los sistemas inteligentes. No hay una clara división entre la percepción (abstracción) y el razonamiento en el mundo real.

En los 80's, los investigadores empezaron a buscar alternativas al paradigma de la IA simbólica, para tratar problemas relacionados con la operación eficiente de sistemas con restricciones temporales, y la representación y el razonamiento sobre entornos complejos, dinámicos y físicos. Las nuevas aproximaciones rechazan las representaciones simbólicas y la toma de decisiones basada en la manipulación de estas representaciones; insisten en la idea que el comportamiento racional inteligente está innatamente ligado con el entorno que el agente ocupa, que el comportamiento inteligente no es etéreo sino que es producto de la interacción que mantiene el agente con su entorno; y también, que el comportamiento inteligente emerge de la interacción de varios comportamientos simples [12].

Para Brooks [2] y [3], existen 4 puntos claves en robótica, que han llevado a esta aproximación:

- **Situadness:** Los robots están situados en el entorno, no manejan descripciones abstractas, sino que el entorno influencia directamente el comportamiento del sistema.
- **Embodiment:** Los robots tienen cuerpo y padecen el mundo directamente, sus acciones son parte de la dinámica con el mundo, y tienen efectos inmediatos en las propias sensaciones del robot.
- **Inteligencia:** La inteligencia del robot proviene también, la situación del entorno, la transformación de las señales en los sensores y del acoplamiento físico del robot con el entorno.

- **Emergence:** La inteligencia del sistema emerge de las interacciones de éste con el entorno y algunas veces, de las interacciones indirectas entre sus componentes.

Un aporte importante lo han hecho Zhang y Mackworth [15], quienes han definido a un robot como un sistema híbrido, inteligente y dinámico consistente en un controlador acoplado al cuerpo. Así mismo, el robot está simétricamente unido al entorno. Para modelar la dinámica del sistema y del entorno, han desarrollado lo que se denomina Constraint Net (CN), un soporte unitario para modelar sistemas dinámicos. CN puede tratar sistemas dinámicos y discretos tanto si son estructuras de eventos sincrónicos como asincrónicos. Está implementado como módulos con puertos de entrada/salida (I/O). Cada módulo realiza una transducción de las entradas a las salidas, sujetas al principio de causalidad: una salida en cualquier momento puede depender sólo de los valores anteriores y actuales de las entradas. Un robot situado puede modelarse separadamente como tres máquinas:

- El cuerpo del robot.
- El controlador del robot.
- El entorno.

Teniendo un modelo CN y las propiedades requeridas del controlador, especificadas como un conjunto de restricciones, es factible (si es posible) encontrar de forma automática, un controlador con las especificaciones propuestas [14].

De acuerdo con Asada [1], los cuerpos físicos juegan un papel importante en la consecución de los objetivos de los agentes en un mundo dinámico real. Le permiten al sistema, interactuar con entornos físicos, que hacen que el sistema aprenda del entorno y desarrolle su representación interna. El significado de tener un cuerpo físico es:

- Las capacidades sensoriales y las capacidades actuadoras están altamente relacionadas (no se pueden separar).
- El agente decide de acuerdo con sus planes; los espacios sensoriales y actuadores tienen que abstraerse, ya que están limitados por los recursos que lleva (memoria de la CPU, velocidad de la CPU, controladores, etc.).

- Esta abstracción depende tanto de la personificación fundamental del agente, como de las experiencias (interacciones que el agente ha experimentado con su entorno).
- La consecuencia de esta abstracción es una representación subjetiva del entorno, su evaluación se hace por medio de los resultados de los comportamientos.
- En el mundo real, las interacciones entre agentes, y agentes y ambiente son asíncronas, complejas y concurrentes.
- Las interacciones físicas reales dan buenos datos para el aprendizaje (los agentes software carecen de esto).

3.3 Arquitectura DPAA.

Los agentes físicos autónomos que componen un sistema multi-agente necesitan cumplir ciertos requerimientos para poder trabajar en el mundo real, en tiempo real. Estos requerimientos deben estar implícitos en la propia arquitectura del agente. Oller, [9] destaca cuatro de ellos que son:

- **Comportamiento situado:** Los agentes deben saber reconocer los eventos asíncronos y reaccionar a tiempo y de forma apropiada teniendo en cuenta su cuerpo físico.
- **Comportamiento hacia el objetivo:** Los agentes deben seleccionar las acciones basándose en lo que quieren lograr a escala individual y colectiva.
- **Eficiencia:** Las tareas se tienen que ejecutar eficientemente, teniendo en cuenta las posibilidades físicas reales (temporales y espaciales) que tienen para conseguirlo.
- **Coordinación:** Los agentes deben tener presente las interacciones positivas y negativas con los demás agentes.

Para cubrir estos aspectos, Oller propone una arquitectura agente como la mostrada en la Figura 3-4. Ésta propone el uso de tres módulos especializados que son:

- **Módulo de control:** es la conexión directa del agente con el mundo real.
- **Módulo supervisor:** es la interfaz entre los parámetros del mundo real con el mundo lógico del agente.
- **Módulo agente:** es la conexión con el mundo multi-agente.

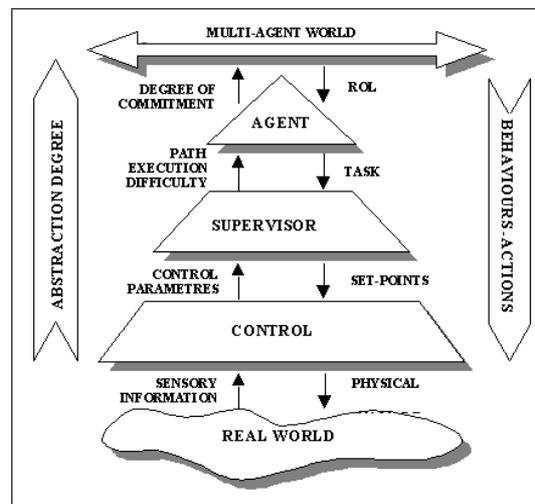


Figura 3-4: Arquitectura del agente.

A medida que aumenta el grado de abstracción, la naturaleza de los comportamientos es más sofisticada y tiende a ser directamente independiente del mundo real.

A la hora de tomar una decisión respecto a una tarea a hacer, el agente, después de negociar con los demás, se basa tanto en parámetros externos del entorno como en los estados internos propios. Los parámetros externos pueden obtenerse a través de intercambio de información entre agentes. Por otro lado, los internos deben describir los estados del cuerpo físico del agente, tanto a alto como a bajo nivel.

En esta arquitectura se propone incluir estos parámetros internos dentro de las capacidades del agente, y dependiendo del grado de abstracción de la información se definen tres tipos de capacidades diferentes:

- **Capacidades atómicas:** Consisten en ser la percepción de las cualidades de un controlador. Integran la identificación del cuerpo físico del agente, la percepción del entorno a través del cuerpo y, en última instancia, la adaptación del agente al entorno (aprendizaje).
- **Capacidades básicas:** Se obtienen combinando las capacidades atómicas y representan el conocimiento del supervisor. En éstas, comienza a utilizarse un nivel de abstracción que no es completamente simbólico, pero tampoco completamente numérico.
- **Capacidades simbólicas:** Se definen de manera simbólica, ya que contienen información del modelo del mundo (abstracto). Dependen de la aplicación para la que se ha creado el agente.

Antes de tomar una decisión, el agente consulta su conjunto de capacidades. Si las capacidades contienen suficiente información respecto de lo que quiere hacer, el agente aceptará o rechazará la acción propuesta. En caso de no ser suficiente, el agente se comunicará con el módulo inmediatamente inferior (y éste con el siguiente) para obtener la información que necesita. Así, al aceptar una tarea determinada, el agente es consciente de la acción que su cuerpo le permite realizar con un grado de certeza elevado.

3.4 Resumen

La Inteligencia Artificial (IA) se inicia formalmente en 1956 cuando se acuñó el término, no obstante, ya había investigadores trabajando en ella desde hacía 5 años. Comenzó como un campo cuyo objetivo era replicar el comportamiento humano en máquinas.

Los primeros trabajos de la IA se concentraron en juegos, problemas geométricos, álgebra simbólica, prueba de teoremas y otra clase de sistemas formales. En cada caso, la semántica de los dominios era bastante simple.

Al final de los 60's y comienzo de los 70's, la técnica de representar al mundo con bloques se hizo popular entre los investigadores de IA. Tenía una semántica uniforme y simple. La clave de su éxito consistió en representar el estado del mundo completa y explícitamente. Las

técnicas de búsqueda podían utilizarse para hacer planificaciones dentro de este mundo bien conocido. También se podía aprender. Eventualmente, la crítica sacó a relucir que los bloques eran un “mundo de juguetes” y que dentro de él, había simplemente soluciones con propósitos especiales a los que debían considerarse como problemas más generales. Pronto apareció un nuevo eslogan: “La buena representación es clave para la IA”. La idea era que representando explícitamente sólo los hechos pertinentes, la semántica del mundo se reduciría a un simple sistema cerrado otra vez. La abstracción aplicada a los detalles importantes solamente, simplificaba los problemas.

La única entrada en la mayoría de los programas de IA está restringida a un conjunto de afirmaciones deducidas de la realidad por humanos. Los problemas de reconocimiento, comprensión espacial, tratamiento del ruido en los sensores, modelos parciales, etc. se ignoran. Estos problemas se relegan al dominio de entradas de cajas negras. Evidencias psicofísicas sugieren que todos están íntimamente ligados con la representación del mundo usada por los sistemas inteligentes. No hay una clara división entre la percepción (abstracción) y el razonamiento en el mundo real.

En los 80's, los investigadores empezaron a buscar alternativas al paradigma de la IA simbólica, para tratar problemas relacionados con la operación eficiente de sistemas con restricciones temporales, y la representación y el razonamiento sobre entornos complejos, dinámicos y físicos. Las nuevas aproximaciones rechazan las representaciones simbólicas y la toma de decisiones basada en la manipulación de estas representaciones; insisten en la idea que el comportamiento racional inteligente está innatamente ligado con el entorno que el agente ocupa, que el comportamiento inteligente no es etéreo sino que es producto de la interacción que mantiene el agente con su entorno; y también, que el comportamiento inteligente emerge de la interacción de varios comportamientos simples.

Esta visión hace resurgir a los sistemas multi-agentes (MAS) cuyo objetivo es investigar el comportamiento de un conjunto de agentes autónomos (posiblemente preexistentes), que trata de resolver un problema determinado. En los MAS, los agentes tienen un grado mayor de autonomía y pueden decidir dinámicamente qué interacciones son adecuadas, qué tareas deben realizar y quién las realiza. Además, es posible mantener un conocimiento que no es globalmente consistente e, incluso, los agentes pueden tener objetivos globales diferentes.

Algunas razones para incrementar el interés en la investigación de los MAS incluye: la habilidad de proporcionar robustez y eficiencia; la habilidad para permitir la operación entre sistemas heredados existentes; y la habilidad para resolver problemas en los que los datos, la experticia o el control están distribuidos.

Un *agente*, según Wooldridge [12], es un sistema computacional que está *situado* en algún *entorno*, y que es capaz de realizar *acciones autónomas* en su entorno con el propósito de conseguir los objetivos para los que fue diseñado. Realizar *acciones autónomas* significa que los agentes operan sin la intervención directa de humanos u otros sistemas, y tienen el control sobre sus estados internos y sobre su comportamiento.

Normalmente un agente posee un repertorio de acciones que representan la capacidad de efecto (effectoric capacity), es decir, su habilidad para modificar su entorno. Como no todas las acciones pueden ejecutarse siempre, existen unas precondiciones (pre-conditions) asociadas a ellas, que definen las posibles situaciones en las cuales pueden aplicarse.

Actualmente, se intentan desarrollar robots inteligentes combinando las técnicas de la IA con las teorías tradicionales de control, pero, aún no se ha podido definir una interfaz entre estas áreas y tampoco coordinar ambos niveles, ni analizar el comportamiento del sistema completo.

Tanto como Brooks [2] como Mackworth [13] insisten en que la inteligencia de un sistema se determina por la dinámica de interacción con el mundo.

Brooks ha desarrollado la *arquitectura subsumption*, el mejor ejemplo de arquitectura reactiva. En este tipo de arquitectura, la toma de decisión del agente se realiza a través de un conjunto de comportamientos que cumplen tareas; cada comportamiento puede pensarse como una función, que toma continuamente las entradas de percepción y las correlaciona con acciones a realizar. Cada uno de estos módulos de comportamientos se diseñan para conseguir alguna tarea en particular. Un punto importante a destacar es que, estos módulos no incluyen representaciones simbólicas complejas ni hacen ningún tipo de razonamiento simbólico. En esta arquitectura, lo difícil es obtener un comportamiento global resultante de la interacción entre los diferentes comportamientos integrantes. Asimismo, esta interacción entre comportamientos puede resultar compleja, ya que cada vez que se agrega un nuevo comportamiento, es necesario tener en cuenta también, la relación entre las competencias de

los comportamientos al momento de decidir cuál toma el control. Por otra parte, hay que conocer todas las posibles situaciones que puedan presentarse y crear los comportamientos adecuados.

Paralelamente, Zhang y Mackworth [15] han desarrollado un soporte unitario para modelar sistemas dinámicos denominado Constraint Net (CN). Ellos definen a un robot como un sistema híbrido, inteligente y dinámico que tiene un controlador acoplado al cuerpo y que está simétricamente unido al entorno. Está implementado como módulos con puertos de entrada/salida (I/O). Cada módulo realiza una transducción de las entradas a las salidas, sujetas al principio de causalidad: una salida en cualquier momento puede depender sólo de los valores anteriores y actuales de las entradas. Un robot situado puede modelarse separadamente como tres máquinas que son el cuerpo del robot, el controlador del robot y el entorno. Teniendo un modelo CN y las propiedades requeridas del controlador, especificadas como un conjunto de restricciones, es factible (si es posible) encontrar de forma automática, un controlador con las especificaciones propuestas [14]. CN tiene varios niveles de abstracción. Con una formalización rigurosa, CN provee de una semántica de programación para el diseño de sistemas de control de robots. Es una aproximación formal para modelar sistemas robóticos, pero no se han presentado resultados prácticos que usen esta arquitectura.

La arquitectura DPAA desarrollada por Oller, ha sido pensada para representar agentes físicos. Es una arquitectura BDI en capas que incluye comportamientos reactivos y cuyo proceso de decisión se basa en la lógica (deliberación). Aprovecha la estructura y las instrucciones del lenguaje de programación AGENTO, y plantea utilizar las capacidades del agente para contener el conocimiento de la dinámica de su cuerpo, por lo que define tres tipos de capacidades diferentes. Estos tipos de capacidades están relacionados con el grado de abstracción que tiene la información que contienen y con los diferentes módulos de la estructura. Las capacidades atómicas engloban el conocimiento que tiene el agente sobre su cuerpo físico (parámetros del mundo real, no hay información simbólica) y están asociadas con el módulo de control. Las capacidades básicas resultan de la combinación de las capacidades atómicas y en ellas coexisten parámetros relativos al mundo real y al mundo simbólico, relacionados con el lenguaje agente. Se vinculan al módulo supervisor. Las capacidades simbólicas contienen la información relacionada con el modelo del mundo (conocimiento abstracto), dependen de la aplicación para la que se ha desarrollado el agente, y

están ligadas al módulo agente. Cuando el agente debe decidir que acción realizar, una vez realizada la negociación con los demás agentes, se basa tanto en atributos externos (vinculados con el entorno) como internos (descripción del cuerpo físico del agente). Esta aproximación tiene el problema que los parámetros definidos para representar la dinámica del cuerpo físico del agente, son deficientes y no la especifican completamente.

Zhang y Mackworth en [13] mencionan que la tendencia actual para desarrollar sistemas físicos inteligentes, es combinar las técnicas de la Inteligencia Artificial con la teoría de control tradicional, pero hacen hincapié en la falta de una interfaz entre la parte de alto nivel (AI) y la de bajo nivel (control).

Analizando las diferentes posibilidades, se cree que la arquitectura DPAA parece ser la más adecuada para adaptarse a este trabajo, ya que permite incluir a priori, conocimiento respecto del cuerpo físico del agente y tenerlo en cuenta al momento de tomar una decisión lógica; con lo que se estaría estableciendo unos lazos entre los niveles mencionados anteriormente. Además, permite ampliar las capacidades a medida que el agente va aprendiendo que puede hacer con su cuerpo, de manera que finalmente se podría tener un modelo computacional on-line tanto en el ámbito de control (normal en estos sistemas) como en el de la IA (anormal ya que son off-line).

4 Objetivos.

Los objetivos principales de este trabajo son representar la dinámica del cuerpo físico de un agente en las capacidades atómicas y esbozar la manera de utilizar este conocimiento en su algoritmo de decisión, de modo tal que, al adquirir compromisos con sus semejantes, éste sepa con “seguridad” que las acciones propuestas son físicamente realizables.

Para conseguirlos, primero es necesario cumplir con una serie de metas parciales, que son:

- 1. Analizar los sistemas SISO para determinar qué características son importantes para representar la respuesta dinámica de cualquier sistema.**

Basados en la teoría de control clásica, se eligen los parámetros que mejor representen la respuesta transitoria de cualquier sistema de una entrada una salida (sistemas SISO), teniendo en cuenta las especificaciones de control y el rendimiento del controlador.

- 2. Definir atributos generales, que relacionados con las características mencionadas en el punto anterior, formarán las capacidades atómicas.**

Teniendo en cuenta las características anteriores, se define un conjunto de atributos que será computacionalmente entendibles por el agente. Estos atributos se plantearán de manera que el conocimiento que contengan sea suficientemente general y que las capacidades atómicas puedan compararse entre sí.

- 3. Plantear el uso del conocimiento de la dinámica en el razonamiento del agente.**

Una vez representada la dinámica del cuerpo físico del agente en las capacidades atómicas, hay que buscar la manera de aprovechar ese conocimiento, para que el agente tome decisiones físicamente realizables. Dentro de los límites de este trabajo, se pretende esbozar una manera de utilizar las capacidades.

5 Capacidades Atómicas.

A continuación se describirán algunas de las características comunes a la respuesta temporal y frecuencial de un sistema realimentado. Éstas servirán posteriormente para definir los atributos que formarán las capacidades atómicas.

5.1 Teoría de control.

La respuesta en el tiempo de un sistema de control se divide normalmente en dos partes: la respuesta transitoria y la respuesta en estado estable. Todos los sistemas de control estables reales presentan un fenómeno transitorio antes de alcanzar la respuesta de estado estable y frente a los cambios súbitos en la entrada que no pueden seguir en forma instantánea. En consecuencia, la respuesta transitoria de un sistema de control es necesariamente importante, ya que es una parte significativa del comportamiento dinámico del sistema y la desviación entre la respuesta de la salida y la entrada, o respuesta deseada, se debe controlar cuidadosamente antes de alcanzar el estado estable.

La respuesta de estado estable de un sistema de control también es muy importante, ya que indica en dónde termina la salida del sistema cuando el tiempo se hace grande. Si la respuesta de estado estable no concuerda exactamente con la referencia deseada, se dice que el sistema tiene un **error de estado estable**.

Con frecuencia, las características de desempeño de un sistema de control se especifican en términos de la respuesta temporal para una entrada escalón unitario, dado que ésta es fácil de generar y suficientemente drástica.

El desempeño de un sistema realimentado se describe generalmente, en términos de estabilidad, sensibilidad, precisión y la respuesta transitoria.

5.1.1 Estabilidad Relativa.

La estabilidad relativa es una propiedad básica que es esencial en la aplicación de control automático. El sistema de control debe asegurar que la variable de salida alcanzará un intervalo alrededor del punto de trabajo. Si, ante un cambio de la dinámica del proceso o perturbaciones actuando sobre el mismo, la variable de salida se mueve del punto de trabajo, el sistema de control debe devolverla a la banda requerida, aplicando una acción de control que sea compatible con las restricciones físicas del proceso y de sus actuadores.

Un sistema de control realimentado debe ser estable aún cuando el sistema esté sujeto a señales de control, a entradas extrañas en cualquier parte del lazo, variaciones en la fuente de alimentación y cambios en los parámetros del lazo. Lo importante es cuán estable es el sistema.

La estabilidad relativa del sistema se puede determinar de diversas formas: evaluando el margen de fase y el margen de ganancia, en el análisis frecuencial o el sobrepaso máximo en el análisis temporal.

5.1.2 Estabilidad absoluta.

La estabilidad absoluta de un sistema lineal en lazo cerrado se determina a partir de los polos de lazo cerrado en el plano s . Si alguno de los polos se encuentra en el semiplano derecho de s , entonces el sistema es inestable. Si los polos son complejos conjugados y están sobre el eje $j\omega$, entonces el sistema es oscilatorio puro pero el ruido hace al sistema inestable. Si todos los polos en lazo cerrado se encuentran en el semiplano izquierdo de s , entonces el sistema es estable. La estabilidad absoluta de un sistema es una propiedad del sistema mismo.

5.1.3 Robustez.

Un lazo de control “robusto” es aquel que funciona bien incluso en presencia de cambios moderados en los parámetros del proceso (ganancia, constante de tiempo, retardo, etc.).

5.1.4 Sensibilidad.

La sensibilidad es una medida de la dependencia de las características de un sistema respecto de un elemento en particular. Se puede calcular como:

$$S_K^H(s) = \frac{dM(s)/M(s)}{dK(s)/K(s)}$$

donde

$$M(s) = \frac{Y(s)}{R(s)} \text{ es la función de transferencia del sistema.}$$

K es el elemento que varía.

Esta definición es válida sólo para cambios pequeños (sistema ideal $S = 0$).

5.1.5 Errores de estado estable.

El que un sistema determinado exhiba un error en estado estable para un tipo específico de entrada depende del tipo de función de transferencia en lazo abierto del sistema.

Los sistemas de control se clasifican de acuerdo con su capacidad de seguir entradas escalón, rampa, parábola, etc. en sistemas de tipo 0,1,2,..., N, dependiendo de la cantidad de integradores puros ($s=0$) que tenga. Conforme el número del tipo es mayor, mejora la precisión sin embargo agrava el problema de la estabilidad relativa. Siempre es necesario un **equilibrio** entre la **precisión** en estado estable y la **estabilidad relativa**.

El error de estado estacionario se calcula como:

$$e_{ss} = \lim_{s \rightarrow 0} \frac{sR(s)}{1 + Gc(s)Gp(s)H(s)}$$

donde

R(s) es la entrada al sistema.

Gc(s) la función de transferencia del controlador

Gp(s) la función de transferencia de la planta.

H(s) la función de transferencia de realimentación.

Aplicando esta definición del error de estado estacionario para las entradas típicas, se obtiene:

- Respuesta escalón:

Para una entrada escalón $R(s) = A/s$ el error de estado estacionario queda:

$$e_{ss} = \frac{A}{1 + \lim_{s \rightarrow 0} Gc(s)Gp(s)H(s)} = \frac{A}{1 + Kp}$$

donde la *constante de error de posición* es:

$$Kp = \lim_{s \rightarrow 0} Gc(s)Gp(s)H(s)$$

Para sistemas de tipo 1 o superior, el error de estado estacionario es cero. Para sistemas tipo cero, es una constante distinta de cero, cuyo valor queda definido por la ecuación anterior.

- Respuesta rampa:

Para una entrada rampa $R(s) = A/s^2$ el error de estado estacionario queda:

$$e_{ss} = \frac{A}{\lim_{s \rightarrow 0} sGc(s)Gp(s)H(s)} = \frac{A}{Kv}$$

donde la *constante de error de velocidad* es:

$$Kv = \lim_{s \rightarrow 0} sGc(s)Gp(s)H(s)$$

Para sistemas de tipo 2 o superior, el error de estado estacionario es cero. Para, tipos 1, es una constante distinta de cero, cuyo valor queda definido por la ecuación anterior y para sistemas de tipo 0 es infinito.

- Respuesta parabólica:

Para una entrada parabólica $R(s) = A/s^3$ el error de estado estacionario queda:

$$e_{ss} = \frac{A}{\lim_{s \rightarrow 0} s^2 Gc(s)Gp(s)H(s)} = \frac{A}{Ka}$$

donde la *constante de error de aceleración* es:

$$Ka = \lim_{s \rightarrow 0} s^2 Gc(s)Gp(s)H(s)$$

Para sistemas de tipo 3 o superior, el error de estado estacionario es cero. Para, tipos 2, es una constante distinta de cero, cuyo valor queda definido por la ecuación anterior y para sistemas de tipo 0 y 1 es infinito.

5.1.6 Precisión.

Es minimizar el error de estado estacionario para una determinada clase de entradas. Es deseable en un sistema de control que responda con el mínimo error a cambios en la posición, velocidad, aceleración y derivadas de orden superior.

El error depende del número de polos en el origen en el plano complejo, es decir, del *tipo del sistema*.

5.1.7 Rechazo de perturbaciones.

Todos los sistemas físicos tienen entradas indeseadas (ruido en la medida, cargas o descargas, etc.), llamadas perturbaciones, además de las usadas para controlar el sistema. En general se desea que el sistema no responda de manera significativa a éstas. Las perturbaciones son señales que no se pueden controlar, y generalmente se intenta diseñar al sistema de control para que éstas tengan un efecto mínimo.

Existen 4 maneras de minimizar el efecto de las perturbaciones. Éstas son:

- Modificar el diseño de la planta.
- Incrementar la ganancia del lazo aumentando la ganancia del controlador, pero manteniendo la ganancia de la función de transferencia de las perturbaciones sin modificar (difícil).
- Reducir la magnitud de la perturbación.
- Utilizar el método feedforward (siempre que la perturbación pueda medirse con un sensor).

5.1.8 Respuesta transitoria.

La mayoría de las veces, las características de desempeño deseadas de un sistema se especifican en el dominio temporal; para una entrada escalón unitario (fácil de generar y suficientemente drástica) y como condiciones iniciales, que el sistema esté en reposo (la respuesta transitoria depende de las condiciones iniciales del sistema). Al especificar las características de la respuesta transitoria de un sistema controlado es común especificar:

- **Tiempo de retardo, t_d :** es el tiempo requerido para que la respuesta alcance la primera vez la mitad del valor final.
- **Tiempo de subida, t_r :** es el tiempo requerido para que la respuesta pase del 10% al 90%, del 5% al 95% o del 0 al 100% de su valor final.
- **Tiempo de pico, t_p :** es el tiempo requerido para que la respuesta alcance el primer pico del sobrepaso.
- **Sobrepico máximo, M_p :** es el valor pico máximo de la curva de respuesta, medido a partir de la unidad. Se define como:

$$M_p = \frac{c(t_p) - c(\infty)}{c(\infty)} \times 100\%$$

El M_p indica de manera directa la **estabilidad relativa** del sistema.

- **Tiempo de establecimiento, t_s :** es el tiempo que se requiere para que la curva de respuesta alcance un rango alrededor del valor final (2 a 5 %) y permanezca dentro de él.

5.1.9 Margen de fase y margen de ganancia.

Otra manera de estudiar la respuesta de un sistema es la del dominio frecuencial. En este campo, se definen dos márgenes, el de fase y el de ganancia, que proporcionan una medida de la estabilidad del sistema.

El **margen de fase** es la cantidad de atraso de la fase adicional en la frecuencia de cruce de ganancia, requerida para llevar al sistema al borde de la inestabilidad. La frecuencia de cruce de ganancia es la frecuencia en la cual la magnitud de la función de transferencia en lazo abierto es unitaria. Para que el sistema sea estable, el margen de fase debe ser positivo.

El **margen de ganancia** es el recíproco de la magnitud de la función de transferencia en lazo abierto en la frecuencia a la cual el ángulo de fase es -180° . Para que el sistema sea estable, el margen de ganancia expresado en decibeles, tiene que ser positivo.

5.1.10 Índices de desempeño de controladores.

Los índices de desempeño tratan de comparar la calidad de la regulación del controlador y el esfuerzo de control necesario para alcanzarla. Algunos índices son:

a) *Acerca del error de regulación:*

- Integral del valor absoluto del error ($e(t)$):

$$IAE = \int_{t_1}^{t_2} |e(t)| dt$$

- Integral del error cuadrático:

$$ISE = \int_{t_1}^{t_2} [e(t)]^2 dt$$

- Integral del valor absoluto del error ponderado en el tiempo:

$$ITAE = \int_{t_1}^{t_2} t |e(t)| dt$$

- Variancia del error:

$$Var(e) = \frac{1}{t_2 - t_1} \int_{t_1}^{t_2} [e(t) - E(e)]^2 dt$$

- Raíz cuadrada de la media del error:

$$RMS(e) = \sqrt{\frac{1}{t_2 - t_1} \int_{t_1}^{t_2} [e(t)]^2 dt}$$

b) *Acerca de la señal de control:*

- Integral del valor absoluto de la derivada de la señal de control (u(t)):

$$IADU = \int_{t_1}^{t_2} \left| \frac{du(t)}{dt} \right| dt$$

- Integral de la señal de control cuadrática:

$$ISU = \int_{t_1}^{t_2} [u(t)]^2 dt$$

- Variancia de la señal de control:

$$Var(u) = \frac{1}{t_2 - t_1} \int_{t_1}^{t_2} [u(t) - E(u)]^2 dt$$

- Raíz cuadrada de la media de la señal de control:

$$RMS(u) = \sqrt{\frac{1}{t_2 - t_1} \int_{t_1}^{t_2} [u(t)]^2 dt}$$

5.2 Definición de los Atributos.

Según la arquitectura DPAA existen tres tipos de capacidades. Las más importantes para el desarrollo del presente trabajo son las capacidades atómicas, que por definición contienen la información referida a la dinámica del cuerpo físico del agente. Pero, ¿qué información deben contener? ¿Qué parámetros, computacionalmente entendibles por el agente, son los adecuados para representar la dinámica de su cuerpo físico?

El objetivo es conseguir un conjunto de “fichas técnicas”, que tengan el conocimiento dinámico necesario para que, cuando el agente tenga que decidir, sepa a priori con cierta seguridad, si es físicamente capaz de cumplir con la acción requerida.

Cuando se quiere realizar el control de un sistema, lo primero que se especifica son las condiciones que tiene que cumplir la salida del sistema realimentado. Estos requisitos, describen la respuesta del sistema. Los más comunes han sido mencionados en la sección anterior. Para poder realizar diferentes tareas, un sistema puede tener varios controladores.

Al realizar el control de un sistema, se tiene en cuenta el sistema propiamente dicho, el actuador que se vaya a utilizar y el sensor que servirá para realimentar al sistema. La respuesta dinámica se ve influida por estos elementos que cierran el lazo de control.

Las “fichas técnicas” serán las capacidades atómicas del agente, y deberán ser completadas por el Ingeniero de Control al realizar los controladores del sistema.

Las características que contendrán estas capacidades estarán relacionadas con la respuesta temporal del sistema controlado. Además, se agregarán otras relacionadas con parámetros del controlador, del sistema en lazo abierto, del sensor y del actuador, para futuras aplicaciones, pero que para el alcance de este trabajo no tienen ningún significado específico, solamente es información adicional.

Relacionados con la respuesta temporal del sistema, se han seleccionado las siguientes características:

- **Precisión:** Es la capacidad que tiene el sistema controlado de seguir los cambios de consigna.
- **Sobrepico:** Es el máximo valor que alcanza la respuesta transitoria del sistema.
- **Rapidez:** Representa una medida de velocidad en alcanzar la consigna del sistema realimentado respecto del sistema en lazo abierto.
- **Persistencia:** Es la habilidad que tiene el sistema controlado de seguir la consigna cuando está sometido a señales que afectan negativamente el valor de la salida del sistema.
- **Robustez:** Es la capacidad que tiene el sistema de mantener dentro de un rango aceptable el valor de la salida del sistema controlado cuando existen perturbaciones internas (propias del sistema).
- **Agresividad:** Muestra cuan rápido responde el sistema a los cambios de consigna.
- **Esfuerzo de control:** Constituye una medida del gasto de energía por parte del sistema controlado para mantener la salida en el valor de consigna.
- **Coherencia:** Permitirá al agente determinar si el controlador que usa ha sido diseñado para trabajar bajo las especificaciones que él pide.
- **Identificador:** Sirve para identificar el tipo de control que se ha descrito en la capacidad.

La información necesaria del sistema en lazo abierto es:

- **Orden y tipo:** Cantidad de polos del sistema de lazo abierto y cantidad de polos en el origen.

- **Retardo:** Tiempo aproximado que transcurre desde que se aplica una señal distinta a la actual en la entrada y se observa un cambio en la salida.
- **Ganancia:** Relación entre el valor de la señal de salida en estado estacionario y la señal de entrada al sistema.
- **Constante de tiempo:** Es el tiempo que tarda un sistema de primer orden en alcanzar el 63% del valor de la salida. Esta constante mide cuan rápida es la respuesta del sistema.

Los parámetros que parecen adecuados para representar al controlador son:

- **Identificador:** Nombre del controlador. Por ejemplo PID, Predictivo, etc.
- **Tipo de controlador:** Se especificará si es un controlador lineal o no.
- **Estructura de control:** Feedforward, Multivariable, Selectivo, Control Ratio, Cascada, etc.

Para describir al actuador y al sensor, se ha seleccionado las siguientes características:

- **Clase:** Tipo de actuador o sensor (mecánico, eléctrico, químico, etc).
- **Precisión:** Es el margen en el cual el valor de una magnitud proporcionada puede ser errónea.
- **Sensibilidad:** Mínima variación en la entrada que puede detectar el sensor o a la que responde el actuador.
- **Constante de tiempo:** Tiempo que indica cuan rápido responde el actuador o el sensor a los cambios de consigna.
- **Histéresis:** Es la diferencia de valor de una magnitud dependiendo de si ésta se ha alcanzado por un cambio continuado creciente o decreciente.

- **Dependencia con la temperatura:** Es el cambio del valor de una magnitud, cuando el dispositivo se encuentra trabajando a una temperatura distinta de la nominal.
- **Rango de linealidad:** Rango dentro del cual el actuador o el sensor trabaja en su zona lineal.
- **Retardo:** Tiempo que tarda en cambiar la salida cuando existe un cambio en la entrada.
- **Inmunidad al ruido:** Indica cual es la máxima potencia que puede tener una señal de ruido para que el sensor o el actuador no se vea afectado por ésta.

A continuación se procederá a definir la manera de calcular las características antes descritas.

5.2.1 Precisión

La precisión es un parámetro relacionado con el error que tiene el sistema controlado excitado por una rampa de pendiente τ , al cabo de 2τ :

$$e = \lim_{t \rightarrow 2\tau} \frac{\tau * t - y(t)}{\tau * t} * 100 \quad \text{Ec. 5-1}$$

Donde

τ es la constante de tiempo del sistema en lazo abierto.

$\tau * t$ es una entrada rampa.

Y la precisión será:

$$\text{Precisión} = 100 - e \quad \text{Ec. 5-2}$$

Supóngase un sistema realimentado cuya respuesta a una entrada escalón es la mostrada en la Figura 5-1.

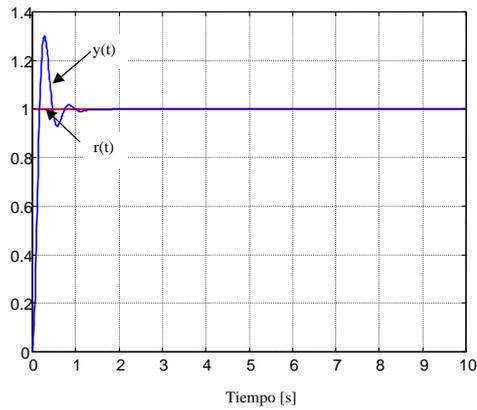


Figura 5-1: Respuesta de un sistema en lazo cerrado.

Sabiendo que el sistema en lazo abierto es uno de segundo orden, cuya función de transferencia es:

$$FT = \frac{1}{s^2 + 3.6s + 9}$$

La constante de tiempo para un sistema de segundo orden es:

$$\tau = \frac{1}{\zeta \omega_n} = \frac{1}{1.8} = 0.556$$

Para calcular la precisión del sistema, hay que generar una consigna rampa

$$r(t) = \tau * t = 0.556t$$

La respuesta del sistema realimentado a la rampa es:

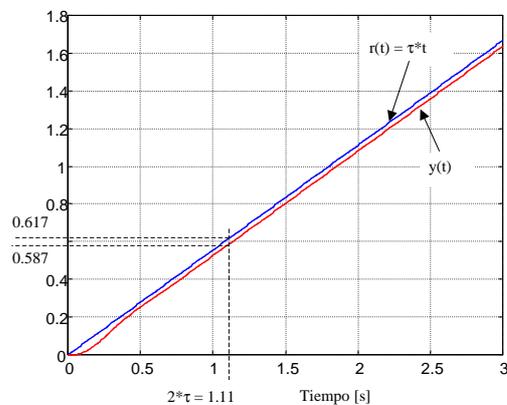


Figura 5-2: Respuesta del sistema controlado a una entrada rampa.

El error se calcula de acuerdo con Ec. 5-1, como:

$$e = \lim_{t \rightarrow 2\tau} \frac{\tau * t - y(t)}{\tau * t} * 100 = \frac{0.617 - 0.587}{0.617} * 100 = 4.86\%$$

Por lo que

$$\text{Precisión} = 100 - e = 100\% - 4.86\% = 95.14\%$$

5.2.2 Sobrepico

El sobrepico máximo de la respuesta transitoria del sistema controlado que se calcula como:

$$M_p = \frac{y(t_p) - y(\infty)}{y(\infty)} \times 100\% \quad \text{Ec. 5-3}$$

Donde

t_p es el tiempo requerido para que la respuesta alcance el primer pico del sobrepico.

Si la respuesta transitoria no presenta sobrepico, el M_p es 0%.

Considerando la respuesta del sistema controlado mostrado en la Figura 5-1 y ampliando la zona del sobrepico máximo:

$$M_p = \frac{1.3 - 1}{1} \times 100\% = 30\%$$

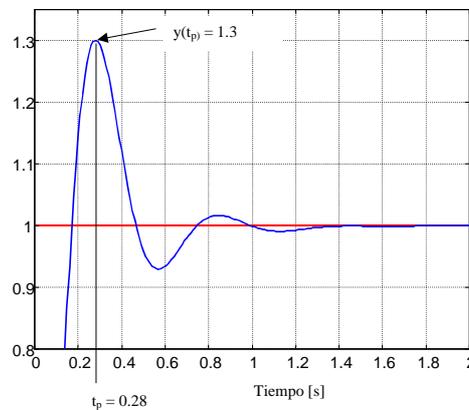


Figura 5-3: Sobrepico máximo.

5.2.3 Rapidez.

La rapidez es un parámetro que representa el tiempo que tarda el sistema controlado en alcanzar el estado estable (tiempo de asentamiento) cuando hay un cambio en la consigna, respecto del tiempo de establecimiento del sistema en lazo abierto. El tiempo de establecimiento o asentamiento (t_s), se define como el tiempo que se requiere para que la curva de respuesta alcance un rango alrededor del valor final (2 a 5 %) y permanezca dentro de él. Por lo tanto la rapidez, es:

$$\text{Rapidez} = \frac{t_{slc}}{t_{sla}} \quad \text{Ec. 5-4}$$

Donde:

t_{slc} : es el tiempo de establecimiento en lazo cerrado.

t_{sla} : es el tiempo de establecimiento en lazo abierto.

Cuanto más pequeño sea este parámetro, más rápida será la respuesta del sistema.

En el ejemplo mostrado en la Figura 5.1, el tiempo de establecimiento es de lazo cerrado es

$$t_{slc} = 0.7s$$

En la Figura 5.4 se muestra la respuesta del sistema en lazo abierto. De ella se puede determinar que el tiempo de establecimiento es:

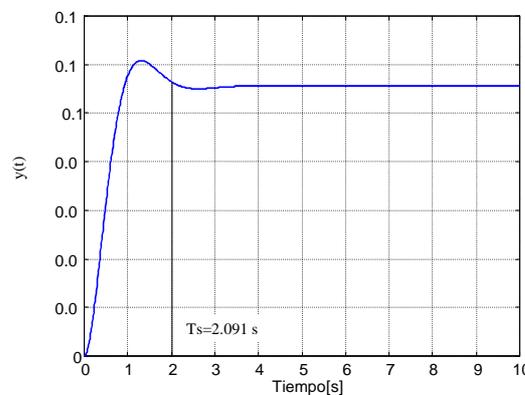


Figura 5.4: Respuesta del sistema en lazo abierto.

$$t_{sla} = 2.091s$$

Por lo que la rapidez es:

$$Rapidez = \frac{t_{slc}}{t_{sla}} = \frac{0.7}{2.091} = 0.3348$$

5.2.4 Persistencia.

Este parámetro está relacionado con la capacidad del sistema controlado para rechazar las perturbaciones que lo puedan afectar, es decir, de mantener la salida dentro de un rango aceptable.

En el caso que la perturbación sea de tipo escalón de amplitud A, la manera de calcular este atributo es:

$$Persistencia = \left[\left(1 - \frac{IAE}{A \times \tau} \right) \times porcentaje_perturbaciones \right] \% \quad \text{Ec. 5-5}$$

Siendo

$$IAE = \int_{t_1}^{t_2} |e(t)| dt$$

τ la constante de tiempo del sistema en lazo abierto.

Se ha elegido la constante de tiempo del sistema en lazo abierto, porque no varía. Si fuese la de lazo cerrado, quedaría determinada por el controlador, y este atributo no sería independiente de él.

En el caso que sea un pulso de área B, este atributo se calcula como:

$$Persistencia = \left[\left(1 - \frac{IAE}{B} \right) \times porcentaje_perturbaciones \right] \% \quad \text{Ec. 5-6}$$

Siendo

$$IAE = \int_{t_1}^{t_2} |e(t)| dt$$

Si $\left(1 - \frac{IAE}{B} \right) < 0$ entonces la persistencia vale 0%, es decir, el sistema es completamente afectado por las perturbaciones (no rechaza ninguna).

Hay que tener en cuenta que el rechazo a perturbaciones es una especificación de diseño del controlador; por lo que, su evaluación queda, en cierta medida, a criterio del ingeniero de control.

En caso de que exista más de un tipo de perturbaciones, este índice valdrá el máximo valor de todas ellas.

Supóngase el sistema controlado de la Figura 5-1. Se sabe que el sistema estará afectado un 90 % de las veces, por perturbaciones de tipo pulso, como el mostrado en la Figura 5-5, aplicadas a la salida del actuador. Como criterio de diseño, se pretende que el controlador las elimine en un tiempo inferior a 3 seg.

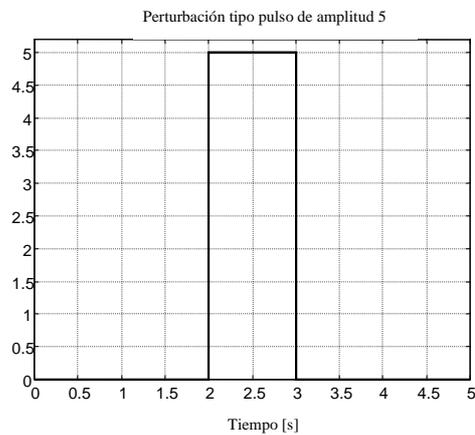


Figura 5-5: Perturbaciones que afectan al sistema.

El área del pulso es

$$B = \text{Amplitud} \times \text{duración} = 5 * 1s = 5s$$

Se considerará a la perturbación rechazada, si la variación de la salida una vez pasada la perturbación es menor al 2 %.

La Figura 5-6 muestra la respuesta del sistema a una perturbación como la presentada en la Figura 5-5.

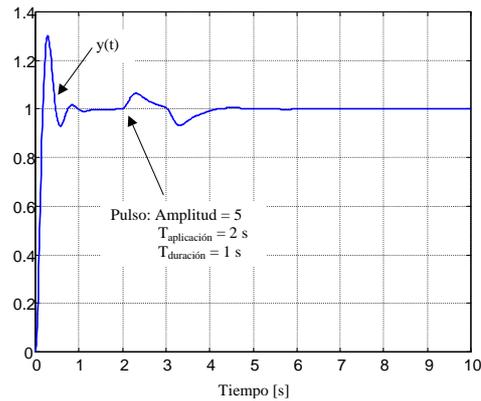


Figura 5-6: Respuesta del sistema controlado a las perturbaciones.

Calculando el IAE desde que se aplica la perturbación hasta que el efecto producido por ésta desaparece, se obtiene:

$$IAE = \int_2^6 |e(t)| dt = 0.0735$$

De la Ec. 5-6

$$Persistencia = \left[\left(1 - \frac{0.0735}{5} \right) \times 90 \right] \% = 88,68\%$$

5.2.5 Robustez.

La robustez es un parámetro que se relaciona con la capacidad del sistema realimentado de mantener la salida dentro de los valores especificados, frente a variaciones de los parámetros del sistema en lazo abierto y a dinámicas no modelizadas.

El margen de fase y el margen de ganancia proporcionan una medida de la estabilidad del sistema. Los márgenes de fase y magnitud proporcionan una medida del cambio máximo que pueden sufrir los parámetros del sistema (ganancia, retardos, dinámicas no modeladas, etc.) y que éste siga siendo estable.

Teniendo en cuenta estos márgenes, y sabiendo cual será la máxima variación de los parámetros que afectan a estas dos variables, se definirá el índice que aquí se denomina **índice de robustez** como:

$$\text{Índice_Robustez} = \frac{\frac{MF}{MF_{nom}} + \frac{MG}{MG_{nom}}}{2} \quad \text{Ec. 5-7}$$

Supóngase el ejemplo representado en la Figura 5-1. El diagrama de Bode del sistema es:

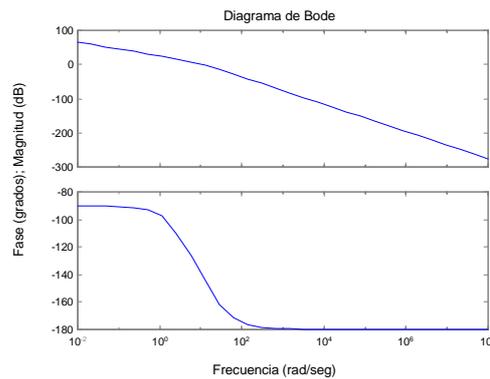


Figura 5-7: Diagrama de Bode del sistema.

A partir de este diagrama, se pueden calcular el margen de fase y el margen de ganancia, que valen:

$$MF_{nom} = 40.3441^\circ$$

$$MG_{nom} = \infty \text{ dB}$$

Sabiendo que existe un retardo no modelado de 0.001 segundos en la función de transferencia del sistema a controlar, se procede a buscar de nuevo los márgenes de fase y de ganancia:

$$MF_{nom} = 39.7203^\circ$$

$$MG_{nom} = 62.7679 \text{ dB}$$

Por lo que el índice de robustez queda (Ec. 5-7):

$$\text{Índice_Robustez} = \frac{\frac{39.7203^\circ}{40.3927^\circ} + \frac{62.7679}{\infty}}{2} = 0.4917$$

5.2.6 Agresividad.

La agresividad es un parámetro que representa la rapidez del sistema para responder a los cambios de consigna. Se definirá como la relación porcentual entre el tiempo de subida (t_r) y el tiempo de establecimiento (t_s) (Ec.5-5).

$$Agresividad = 100\% - \frac{t_r}{t_s} * 100\% \quad \text{Ec. 5-8}$$

Continuando como ejemplo del cálculo de los parámetros de las capacidades atómicas, con el sistema descrito por la Figura 5-1, y se ha ampliado la zona de interés para determinar la agresividad, como se muestra en la Figura 5-8:

La agresividad es:

$$Agresividad = 100\% - \frac{0,17}{0,7} * 100\% = 75.7143\%$$

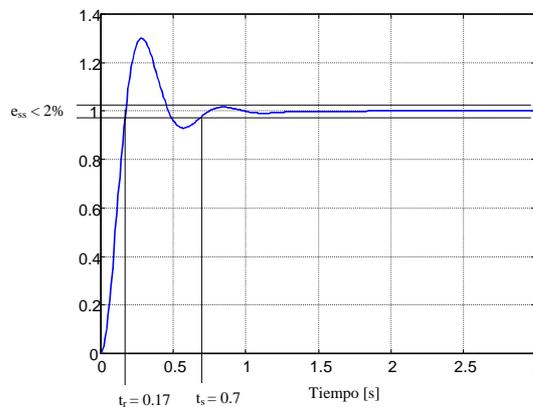


Figura 5-8: Tiempo de subida y de establecimiento.

5.2.7 Esfuerzo de control.

Este parámetro representa el esfuerzo que necesita el controlador para mantener la salida al valor deseado. Para evaluarlo se utilizará el índice IADU, cuya fórmula es:

$$IADU = \int_{t1}^{t2} \left| \frac{du(t)}{dt} \right| dt$$

El esfuerzo de control se definirá como:

$$Esfuerzo_de_control = \frac{IADU}{u_{max} - u_{min}} \quad \text{Ec. 5-9}$$

siendo

u_{max} y u_{min} los valores máximo y mínimo que puede tener la señal de control.

Volviendo al ejemplo de la Figura 5-1, la acción de control necesaria para mantener la salida es la mostrada en la Figura 5-9.

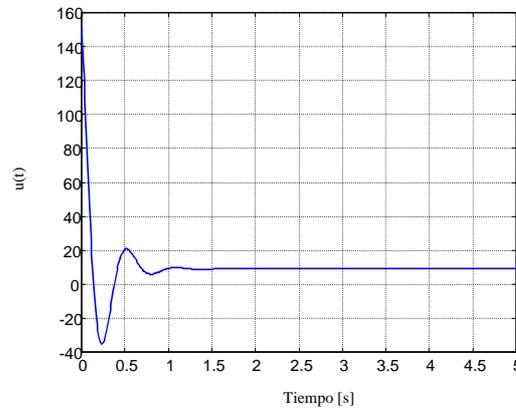


Figura 5-9: Acción de control.

Aplicando la Ec. 5-9, el esfuerzo de control queda:

$$Esfuerzo_de_control = \frac{IADU}{u_{max} - u_{min}} = \frac{263}{160 - (-40)} = 1.3150$$

5.2.8 Coherencia.

La coherencia está relacionada con el rango de trabajo para el cual el controlador diseñado cumple con las especificaciones requeridas.

$$Coherencia = rango_trabajo_sistema \quad \text{Ec. 5-10}$$

Siguiendo con el ejemplo representado en la Figura 5-1, y sabiendo que el sistema realimentado cumple con las especificaciones en el intervalo [0 2], la coherencia valdrá:

$$\text{Coherencia} = [0,2]$$

5.2.9 Identificador del sistema.

El identificador del sistema es un parámetro que permite especificar las unidades de las variables de entrada y salida del sistema controlado, de manera que el agente contraste respuestas comparables del sistema, es decir, si un sistema posee diversos controladores, por ejemplo de velocidad y de posición, al momento de escoger un controlador de velocidad, sólo compare éstos, y no tenga en cuenta los posición.

$$Id = \text{unidad_entrada}, \text{unidad_salida}$$

Si el sistema controlado de la Figura 5-1, tiene por entrada una consigna de velocidad [cm/s] y por salida, también una velocidad, entonces:

$$Id = \text{cm/s}, \text{cm/s}$$

6 Ejemplo.

6.1 Generación de Capacidades Atómicas.

Supóngase un sistema cuya función de transferencia es:

$$FT = \frac{1}{s^2 + 3s + 2}$$

Se pretenden diseñar tres controladores con características diferentes para trabajar en el mismo rango. La Figura 6-1 muestra el diagrama en bloques de la planta más controlador.

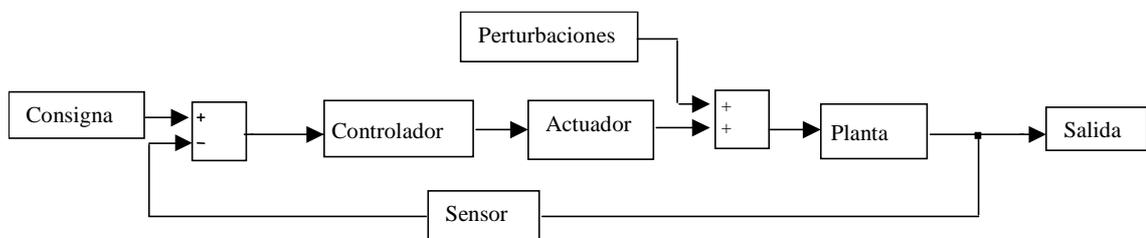


Figura 6-1: Diagrama en bloques del sistema realimentado.

El 95% perturbaciones que afectan al sistema son del tipo pulso amplitud máxima de 10 y de duración de 20 segundos y el 5% restante, de tipo escalón de amplitud 5.

Además el sistema tiene un polo no contenido en el modelo en:

$$\frac{500}{s + 500}$$

Las especificaciones de control que se piden para el sistema realimentado son:

Controlador 1

- Tenga una persistencia al 90%.
- Precisión superior al 95%.

Controlador 2

- Sobrepico inferior al 1%.
- Rechace un 90% de perturbaciones.

Controlador 3

- Tiempo de establecimiento menor a 2s.
- Esfuerzo de control pequeño.

6.1.1 Planta.

La Figura 6-2 muestra la respuesta del sistema, a una consigna escalón de amplitud 3.

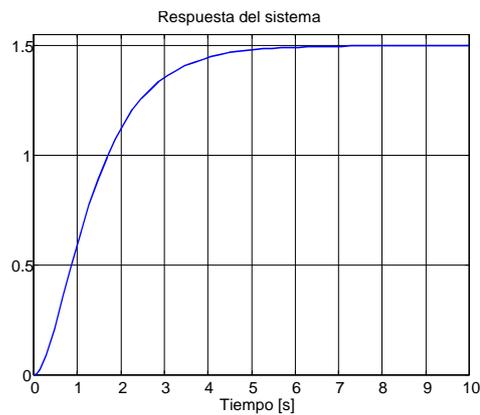


Figura 6-2: Respuesta del sistema en lazo abierto.

La constante de tiempo del sistema es:

$$\tau = \frac{1}{\zeta\omega_n} = \frac{1}{1.5} = 0.6667$$

El tiempo de establecimiento en lazo abierto es:

$$t_{sla} = 4.6s$$

6.1.2 Controlador 1

Teniendo en cuenta las especificaciones de control, se diseña un controlador PID con aproximación de la derivada. La Figura 6-3 muestra la su estructura:

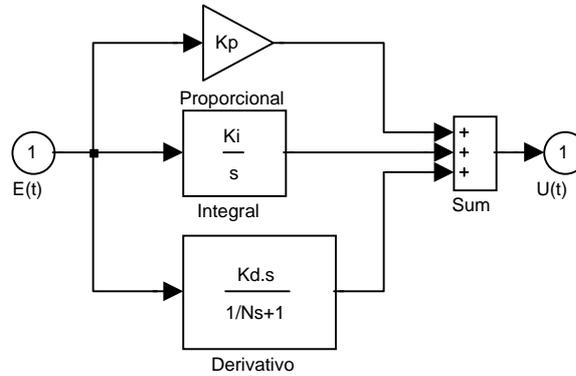


Figura 6-3: Estructura del controlador PID con aproximación de la derivada.

Siendo

$$K_p = 150$$

$$K_i = 40$$

$$K_d = 50$$

$$N = 50$$

La respuesta del sistema controlado es:

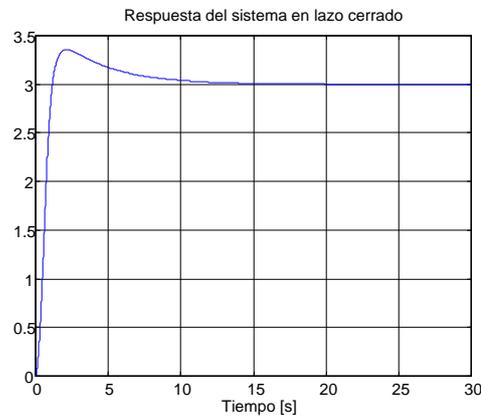


Figura 6-4: Respuesta del sistema controlado con el controlador 1.

Las características de esta respuesta son:

$$M_p = \frac{y(tp) - y(\infty)}{y(\infty)} \cdot 100\% = \frac{3.36 - 3}{3} \cdot 100\% = 12\%$$

$$t_{slc} = 8.6s$$

$$t_r = 0.8945s$$

Por otra parte, la respuesta del sistema controlado para una entrada rampa de pendiente τ es:

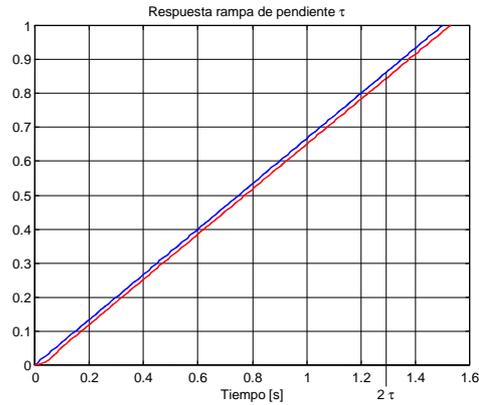


Figura 6-5: Respuesta del sistema controlado a una entrada rampa de pendiente τ ampliada a 2τ

De la Ec. 5-1, el error a 2τ es:

$$e = \lim_{t \rightarrow 2\tau} \frac{\tau^* t - y(t)}{\tau^* t} \cdot 100 = \frac{0.8865 - 0.8687}{0.8865} \cdot 100 = 2.0079$$

El índice IADU es:

$$IADU = \int_{t1}^{t2} \left| \frac{du(t)}{dt} \right| dt = 18.88$$

La respuesta del sistema a las perturbaciones de tipo pulso es:

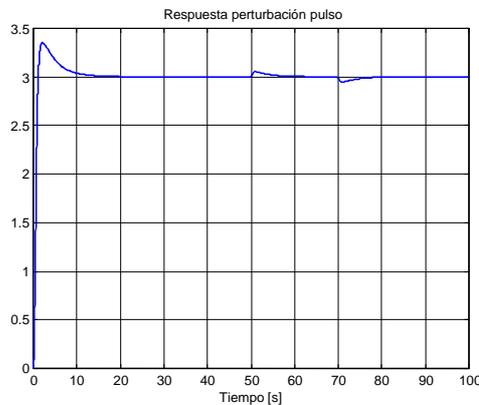


Figura 6-6: Respuesta del sistema realimentado a una perturbación de tipo pulso.

El IAE vale:

$$IAE_{pulso} = \int_{50}^{80} |e(t)| dt = 0.4983$$

$$B = \text{amplitud} \times \text{duración} = 10 \times 20 = 200s$$

Y la respuesta del sistema a perturbaciones escalón de amplitud 5 es:

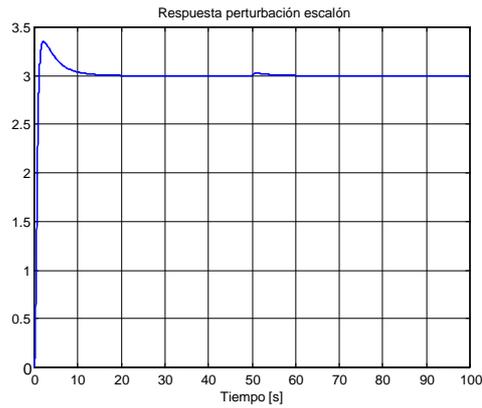


Figura 6-7: Respuesta del sistema realimentado a una perturbación de tipo escalón.

El IAE vale:

$$IAE_{escalon} = \int_{50}^{70} |e(t)| dt = 0.1250$$

El diagrama de Bode del sistema en lazo abierto es:

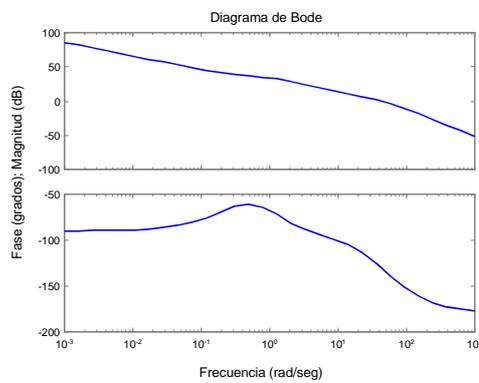


Figura 6-8: Diagrama de Bode del sistema en lazo abierto.

Los márgenes son:

$$MF_{nom} = 51.0080^\circ$$

$$MG_{nom} = 1.8043 \times 10^3 \text{ dB}$$

Considerando el polo no modelado, éstos quedan:

$$MF = 46.3911^\circ$$

$$MG = 10.5363 \text{ dB}$$

Con estos valores se procede a calcular los atributos que forman la capacidad atómica para este controlador.

Teniendo en cuenta la Ec. 5-6

$$Persistencia_pulso = \left[\left(1 - \frac{0.4983}{200} \right) \times 95 \right] \% = 94.7633\%$$

De la Ec. 5-5

$$Persistencia_escalón = \left[\left(1 - \frac{0.1250}{5 \times 0.6667} \right) \times 5 \right] \% = 4.8125\%$$

De la Ec. 5-2

$$Precisión = 100 - 2.0079 = 97.9921\%$$

De la Ec. 5-3

$$Sobrepico = Mp = 12\%$$

De la Ec. 5-4

$$Rapidez = \frac{8.6}{4.6} = 1.8696$$

De las Ec. 5-5 y 5-6

$$Persistencia = \max[persistencia_pulso, persistencia_escalón] = 94.7633\%$$

De la Ec. 5-7

$$Robustez = \frac{\frac{46.3911}{51.0080} + \frac{10.5363}{1.8043 \times 10^3}}{2} = 0.4577$$

De la Ec. 5-8

$$Agresividad = 89.5988\%$$

De la Ec. 5-9

$$Esfuerzo_de_control = \frac{18.88}{15 - (-15)} = 0.6293$$

De la Ec. 5-10

$$Coherencia = [0,6]$$

6.1.3 Controlador 2

Teniendo en cuenta las especificaciones de control, se diseña un controlador PID anti_windup. La Figura 6-9 muestra su estructura.

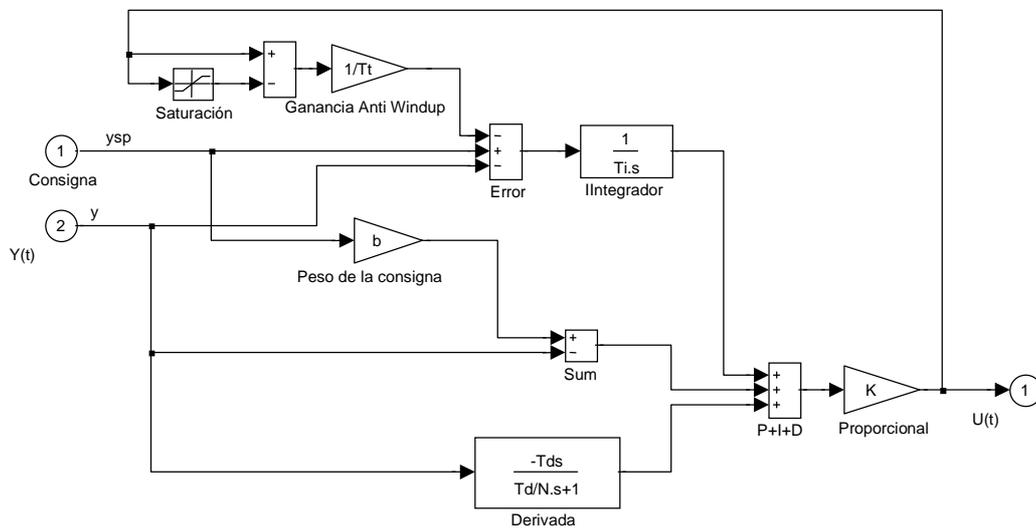


Figura 6-9: Estructura del PID Anti-windup.

Siendo

$$K = 100 \quad Ti = 3.75 \quad Td = 1.25 \quad N = 50 \quad Tt = 0.1 \quad b = 1$$

La respuesta del sistema controlado se muestra en la Figura 6-10.

Las características de esta respuesta son:

$$Mp = \frac{y(tp) - y(\infty)}{y(\infty)} \cdot 100\% = \frac{3.0115 - 3}{3} \cdot 100\% = 0.3833\%$$

$$t_{slc} = 9.4866s$$

$$t_r = 8.0659s$$

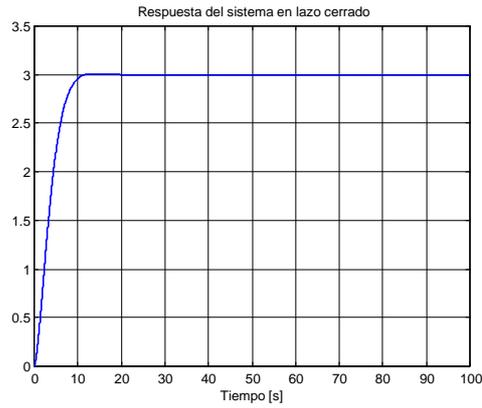


Figura 6-10: Respuesta del sistema en lazo cerrado con el controlador 2.

Por otra parte, la respuesta del sistema controlado para una entrada rampa de pendiente τ es:

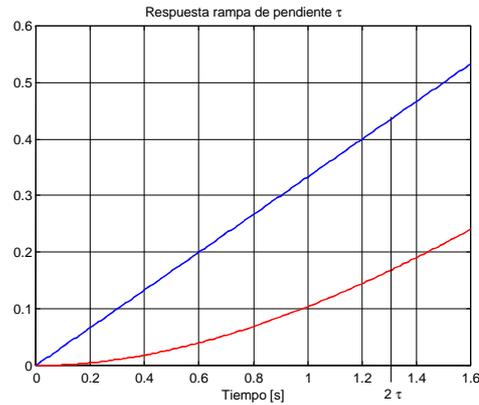


Figura 6-11: Respuesta del sistema controlado a una entrada rampa de pendiente τ ampliada a 2τ

El error a 2τ es, de acuerdo con Ec. 5-1:

$$e = \lim_{t \rightarrow 2\tau} \frac{\tau * t - y(t)}{\tau * t} \cdot 100 = \frac{0.4427 - 0.1737}{0.4427} \cdot 100 = 60.7635\%$$

El índice IADU es:

$$IADU = \int_{t_1}^{t_2} \left| \frac{du(t)}{dt} \right| dt = 64.8635$$

La respuesta del sistema a las perturbaciones de tipo pulso es:

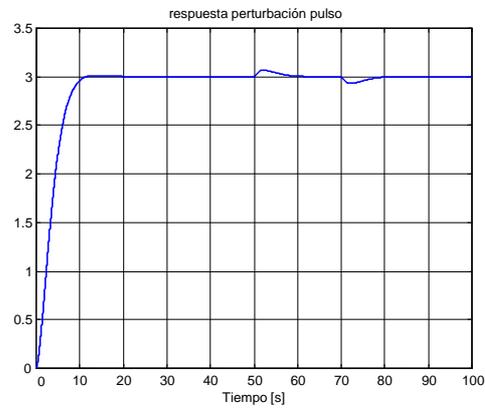


Figura 6-12: Respuesta del sistema realimentado a una perturbación de tipo pulso.

El IAE vale:

$$IAE_{pulso} = \int_{50}^{80} |e(t)| dt = 0.7557$$

$$B = amplitud \times duraci3n = 10 \times 20 = 200s$$

Y la respuesta del sistema a perturbaciones escal3n de amplitud 5 es:

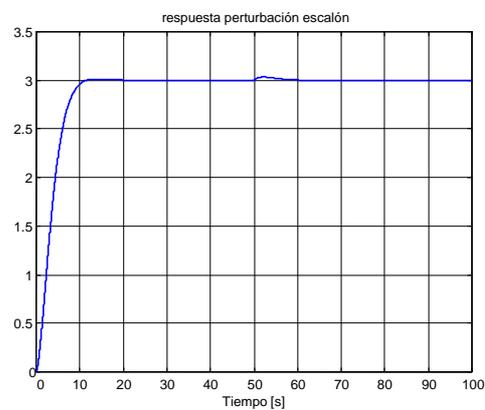


Figura 6-13: Respuesta del sistema realimentado a una perturbaci3n de tipo escal3n.

El IAE vale:

$$IAE_{escalon} = \int_{50}^{70} |e(t)| dt = 0.1889$$

El diagrama de Bode del sistema en lazo abierto es:

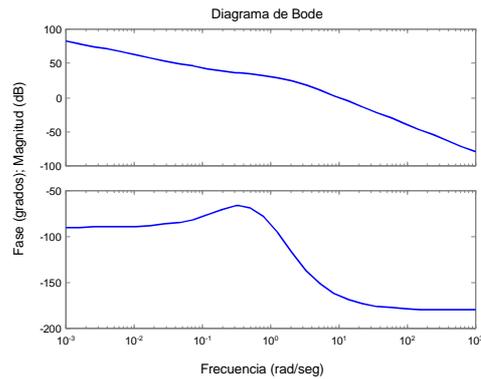


Figura 6-14: Diagrama de Bode del sistema en lazo abierto.

Los márgenes son:

$$MF_{nom} = 16.0276^\circ$$

$$MG_{nom} = 4.7650 \times 10^4 \text{ dB}$$

Considerando el polo no modelado, éstos quedan:

$$MF = 15.6162^\circ$$

$$MG = 13.6207 \text{ dB}$$

Con estos valores se procede a calcular los atributos que forman la capacidad atómica para este controlador.

Teniendo en cuenta la Ec. 5-6

$$Persistencia_{pulso} = \left[\left(1 - \frac{0.7557}{200} \right) \times 95 \right] \% = 94.6410\%$$

De la Ec. 5-5

$$Persistencia_{escalon} = \left[\left(1 - \frac{0.1889}{5 \times 0.6667} \right) \times 5 \right] \% = 4.7166\%$$

De la Ec. 5-2

$$\text{Precisión} = 100 - 60.7635 = 39.2365\%$$

De la Ec. 5-3

$$\text{Sobrepico} = Mp = 0.3833\%$$

De la Ec. 5-4

$$\text{Rapidez} = \frac{9.4866}{4.6} = 2.0623$$

De las Ec. 5-5 y 5-6

$$\text{Persistencia} = \max[\text{persistencia_pulso}, \text{persistencia_escalón}] = 94.6410\%$$

De la Ec. 5-7

$$\text{Robustez} = \frac{15.6162}{16.0276} + \frac{13.6207}{4.7650 \times 10^4} = 0.4873$$

De la Ec. 5-8

$$\text{Agresividad} = 14.9759\%$$

De la Ec. 5-9

$$\text{Esfuerzo_de_control} = \frac{64.8635}{15 - (-15)} = 2.1621$$

De la Ec. 5-10

$$\text{Coherencia} = [0,6]$$

6.1.4 Controlador 3

Teniendo en cuenta las especificaciones de control, se diseña un controlador PID. La Figura 6-15 muestra la estructura del controlador.

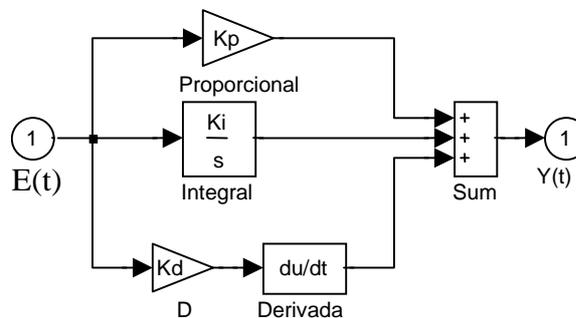


Figura 6-15: Estructura del controlador PID.

Siendo

$$Kp = 3.2594$$

$$Ki = 2.3215$$

$$Kd = 0.3589$$

La respuesta del sistema controlado es:

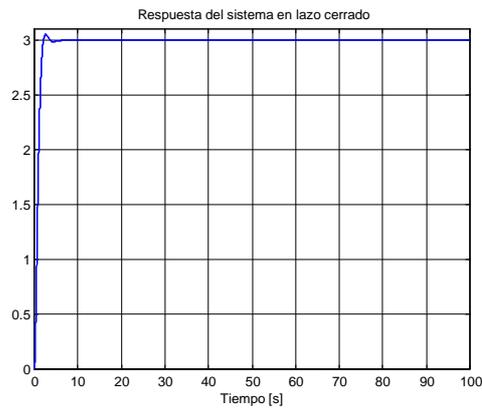


Figura 6-16: Respuesta del sistema realimentado con el controlador 3.

Las características de esta respuesta son:

$$M_p = \frac{y(tp) - y(\infty)}{y(\infty)} \cdot 100\% = \frac{3.0505 - 3}{3} \cdot 100\% = 1.6833\%$$

$$t_{slc} = 1.9143s$$

$$t_r = 1.5495s$$

Por otra parte, la respuesta del sistema controlado para una entrada rampa de pendiente τ es:

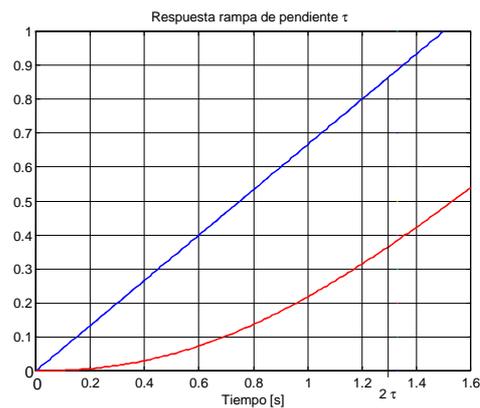


Figura 6-17: Respuesta del sistema controlado a una entrada rampa de pendiente τ ampliada a 2τ

De acuerdo con la Ec. 5-1, el error a 2τ es:

$$e = \lim_{t \rightarrow 2\tau} \frac{\tau * t - y(t)}{\tau * t} \cdot 100 = \frac{0.8858 - 0.3828}{0.8858} \cdot 100 = 56.7848\%$$

El índice IADU es:

$$IADU = \int_{t_1}^{t_2} \left| \frac{du(t)}{dt} \right| dt = 4.4698$$

La respuesta del sistema a las perturbaciones de tipo pulso es:

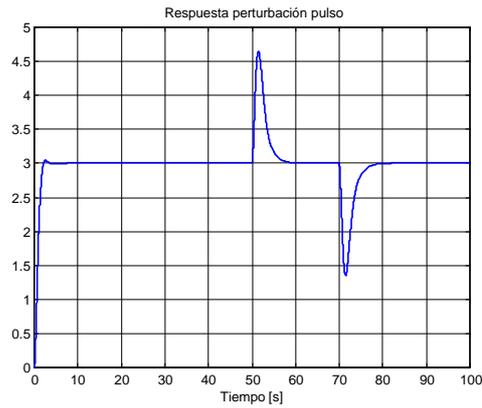


Figura 6-18: Respuesta del sistema realimentado a una perturbación de tipo pulso.

El IAE vale:

$$IAE_{pulso} = \int_{50}^{80} |e(t)| dt = 8.6151$$

$$B = \text{amplitud} \times \text{duración} = 10 \times 20 = 200s$$

Y la respuesta del sistema a perturbaciones escalón de amplitud 5 es:

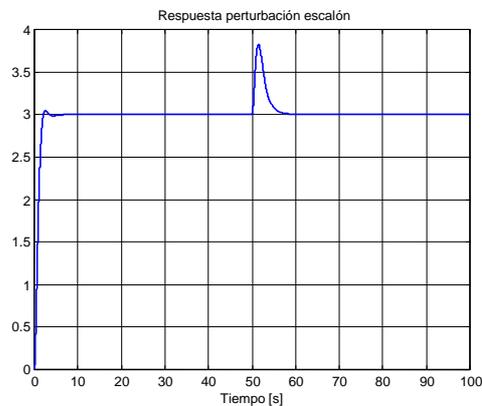


Figura 6-19: Respuesta del sistema realimentado a una perturbación de tipo escalón.

El IAE vale:

$$IAE_{escalon} = \int_{50}^{70} |e(t)| dt = 2.1491$$

El diagrama de Bode del sistema en lazo abierto es:

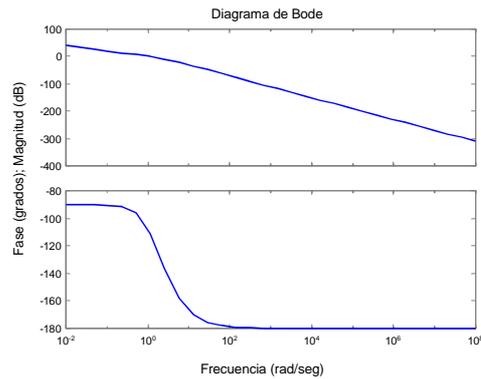


Figura 6-20: Diagrama de Bode del sistema en lazo abierto.

Los márgenes son:

$$MF_{nom} = 67.2301^\circ$$

$$MG_{nom} = \infty \text{ dB}$$

Considerando el polo no modelado, éstos quedan:

$$MF = 67.0858^\circ$$

$$MG = 347.8081 \text{ dB}$$

Con estos valores se procede a calcular los atributos que forman la capacidad atómica para este controlador.

Teniendo en cuenta la Ec. 5-6

$$Persistencia_{pulso} = \left[\left(1 - \frac{8.6151}{200} \right) \times 95 \right] \% = 90.9078\%$$

De la Ec. 5-5

$$Persistencia_{escalon} = \left[\left(1 - \frac{2.1491}{5 \times 0.6667} \right) \times 5 \right] \% = 1.7765\%$$

De la Ec. 5-2

$$\text{Precisión} = 100 - 56.7848 = 43.2152\%$$

De la Ec. 5-3

$$\text{Sobrepico} = Mp = 1.6833\%$$

De la Ec. 5-4

$$\text{Rapidez} = \frac{1.9143}{4.6} = 0.4162$$

De las Ec. 5-5 y 5-6

$$\text{Persistencia} = \max[\text{persistencia_pulso}, \text{persistencia_escalón}] = 90.9078\%$$

De la Ec. 5-7

$$\text{Robustez} = \frac{\frac{67.0858}{67.2301} + \frac{347.8081}{\infty}}{2} = 0.4989$$

De la Ec. 5-8

$$\text{Agresividad} = 19.0566\%$$

De la Ec. 5-9

$$\text{Esfuerzo_de_control} = \frac{4.4698}{15 - (-15)} = 0.1490$$

De la Ec. 5-10

$$\text{Coherencia} = [0,6]$$

La Tabla 6-1 muestra una comparación de todos los atributos de los controladores.

Tabla 6-1: Comparación de los atributos.

Atributos	Controlador 1	Controlador 2	Controlador 3
Precisión	97.9921 %	39.2365 %	43.2152 %
Sobrepico	12 %	0.3833 %	1.6833 %
Rapidez	1.8996	2.0623	0.4162
Persistencia	94.7633 %	94.6140 %	90.9078 %
Robustez	0.4577	0.4873	0.4989
Agresividad	89.5988 %	14.9759 %	19.0566 %
Esfuerzo_de_control	0.6293	2.1621	0.1490
Coherencia	[0, 6]	[0, 6]	[0, 6]

En esta tabla se observa que los diferentes controladores pueden diferenciarse entre sí, y por consiguiente, pueden distinguirse las tres dinámicas distintas conseguidas con ellos.

Cada una de estas dinámicas está representada por una capacidad atómica y éstas son comparables entre sí.

6.2 Utilización de las Capacidades en la Decisión.

En lenguaje AGENT0, los agentes antes de comprometerse con sus pares para realizar una determinada acción, verifican sus creencias y sus capacidades y así se aseguran que podrán cumplir con el compromiso adquirido.

En el caso de agentes físicos, la dinámica de su cuerpo afecta también la correcta realización de la acción, por lo que se han desarrollado las capacidades atómicas, las cuales contienen conocimiento de las diferentes dinámicas conseguidas con los controladores.

Pero, ¿cómo debería tratar este conocimiento el agente? ¿Cómo selecciona el agente la capacidad atómica que más se adecua a la tarea que debe realizar?

A continuación se propone un método de selección de capacidades y por consiguiente, de controlador.

Supóngase que la tarea a realizar exige precisión. En la Tabla 6-1 se observa que el controlador que tiene más precisión es el 1. Sin embargo, si la tarea exige rapidez, el más adecuado es el controlador 3 y si lo importante es que no exista sobrepico, el controlador apropiado es el 2.

¿Qué ocurre cuando la tarea exige precisión y esfuerzo de control mínimo? Por el primer requerimiento se elegiría el controlador 1 mientras que por el segundo, el 3. Sin embargo, el controlador 1 tiene un esfuerzo de control, que sin ser mínimo, no es excesivo y por lo tanto el agente podría escogerlo.

Dependiendo de los requerimientos, el agente deberá ponderar los diferentes atributos de las capacidades y escoger el controlador que más se ajuste a ellos.

¿Y si ahora la tarea exige precisión y muy poco sobrepico? De la misma manera, teniendo en cuenta el primer requisito elegiría el controlador 1 y por el segundo el 2, pero ninguno de los dos cumple con ambas cosas, por lo que el agente debería rechazar el compromiso o al menos, avisar al compañero que la certeza de conseguir el objetivo es muy baja; además debería ponderar de alguna manera que importa más, la precisión o el sobrepico.

Concluyendo las capacidades atómicas condicionan la certeza con la que el agente se compromete a realizar una acción con un compañero.

6.3 Pase en el Fútbol.

Supóngase que los controladores del apartado 6-1 son de posición y se han diseñado para los robots futbolistas; esto significa que el robot posee tres controladores distintos para cubrir una distancia que varía entre 0 y 6 metros.

La Figura 6-21 muestra una situación de juego. El **jugador 1** desea pasar la pelota al **jugador 2** en el punto **p**. La distancia de separación de la pelota del punto **p** es **d_1** mientras que la del jugador 2 al mismo es **d_2** .

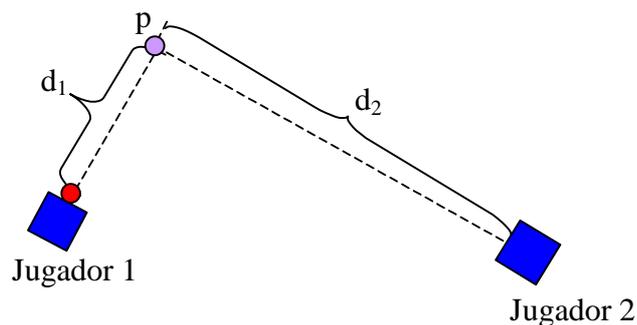


Figura 6-21: Esquema de posición de los jugadores antes de realizar el pase.

Situación 1:

1. **Proposición:** El jugador 1 propone al jugador 2 ir a **p** con precisión.
2. **Introspección:** El jugador 2 busca en su base de capacidades atómicas si tiene un controlador preciso. Encuentra el controlador 1.
3. **Respuesta:** El jugador 2 responde al 1 que puede ser muy preciso al llegar a **p**.
4. **Decisión:** El jugador 1 considera que muy preciso es adecuado y se compromete con 2 para realizar el pase.

Situación 2:

1. **Proposición:** El jugador 1 propone al jugador 2 ir a **p** sin pasarse (sin sobrepico).
2. **Introspección:** El jugador 2 busca en su base de capacidades atómicas si tiene un controlador con poco sobrepico y encuentra el controlador 2.
3. **Respuesta:** El jugador 2 responde al 1 que puede llegar a **p** sin pasarse.
4. **Decisión:** El jugador 1 considera que la respuesta de 2 es buena y se compromete con él para realizar el pase.

Situación 3:

1. **Proposición:** El jugador 1 propone al jugador 2 ir a **p** con precisión y persistencia, ya que hay varios contrarios que pueden obstaculizar a 2.
2. **Introspección:** El jugador 2 busca en su base de capacidades una capacidad que cumpla con estos requerimientos y encuentra la correspondiente al controlador 1.
3. **Respuesta:** El jugador 2 responde al 1 que puede llegar a **p** con precisión y persistencia.
4. **Decisión:** El jugador 1 considera que la respuesta de 2 es buena y se compromete con él para realizar el pase.

Situación 4:

1. **Proposición:** El jugador 1 propone al jugador 2 ir a **p** con precisión y rápidamente.
2. **Introspección:** El jugador 2 busca en su base de capacidades una capacidad que cumpla con estos requerimientos y no encuentra ninguna.
3. **Respuesta:** El jugador 2 responde al 1 que puede llegar a **p** con precisión o rápido.
4. **Decisión:** El jugador 1 considera que es más importante la rapidez que la precisión.
5. **Proposición 2:** El jugador 1 propone a 2 llegar rápido a **p**.
6. **Introspección:** El jugador 2 busca la capacidad atómica que le asegura máxima rapidez.
7. **Respuesta:** El jugador 2 responde a 1 que puede ser muy rápido.

8. **Decisión:** Al jugador 1 le agrada la respuesta y se comprometen al pase.

Situación 5:

1. **Proposición:** El jugador 1 propone al jugador 2 ir a **p** con precisión y rápidamente.
2. **Introspección:** El jugador 2 busca en su base de capacidades una capacidad que cumpla con estos requerimientos y no encuentra ninguna.
3. **Respuesta:** El jugador 2 responde al 1 que puede llegar a **p** con precisión o rápido.
4. **Decisión:** El jugador 1 considera que 2 no puede realizar la acción y no se comprometen a realizar el pase.

Las situaciones anteriores ilustran como sería el algoritmo de decisión de los agentes teniendo en cuenta sus capacidades atómicas.

Cuando los agentes se comprometen, seleccionan la capacidad atómica que asegura el compromiso y utilizan el controlador al que representa para ejecutar la acción.

De la situación 1 a la 3 incluida, los requerimientos que pide el jugador 1 tienen una capacidad atómica asociada directamente, sin embargo la 4 y la 5 no. En la 4 el jugador 1 decide que importa más la rapidez que la precisión, acción para la cual el jugador 2 tiene una capacidad atómica y en la 5, como el jugador 1 exige que se cumplan ambas condiciones y el jugador 2 no tiene ninguna capacidad atómica que pueda asegurar ambos requerimientos, el jugador 1 cancela la proposición (no se comprometen).

En este algoritmo no se han considerado los otros dos tipos de capacidades (simbólicas y básicas) existentes en la arquitectura DPAA, que deberían incluirse en el algoritmo de decisión del agente.

Tampoco se ha especificado que cuando el agente decide, asocia una certeza a los requerimientos de una acción. La certeza es un coeficiente que mide la confianza que tiene el agente sobre la acción a realizar. Ésta se agrega porque las capacidades atómicas son información del cuerpo físico del agente adquiridas a priori, lo que no garantiza que durante la ejecución de la acción, las condiciones sean las mismas que cuando se diseñó el controlador. Asimismo, podría existir información respecto del entorno que condicionase al cuerpo físico del agente y que no estuviese contemplada en las capacidades atómicas.

7 Conclusiones

Los objetivos principales de este trabajo eran representar la dinámica del cuerpo físico de un agente en las capacidades atómicas y esbozar la manera de utilizar este conocimiento en su algoritmo de decisión, mientras que los parciales eran:

- Analizar los sistemas SISO para determinar qué características son importantes para representar la respuesta dinámica de cualquier sistema.
- Definir atributos generales, que relacionados con las características mencionadas en el punto anterior, formarán las capacidades atómicas.
- Plantear el uso del conocimiento de la dinámica en el razonamiento del agente.

El primer objetivo parcial se ha cumplido, creyendo que las características más comunes utilizadas para describir la respuesta dinámica de un sistema SISO real son las descritas en el apartado 5.1.

El segundo objetivo se ha conseguido parcialmente. Las definiciones de los atributos aquí presentadas son una primera aproximación y por lo tanto pueden cambiar cuando se diseñe el algoritmo de decisión. Además como se han especificado considerando solamente sistemas de una entrada – una salida (SISO), puede ocurrir que haya que extender el número de ellos o redefinirlos cuando el estudio abarque sistemas de múltiple entrada – múltiple salida (MIMO). Esta extensión parece posible aunque no inmediata. Incluir los sistemas MIMO es para darle a esta representación una utilidad general.

Respecto al tercer objetivo, todavía queda mucho trabajo que hacer en este campo. La mayoría de los esfuerzos de este trabajo se han concentrado en definir adecuadamente las capacidades, y se ha dejado como trabajo futuro el implementar el algoritmo de decisión. No obstante algunas ideas se han explicado en los apartados 6.2 y 6.3.

La arquitectura DPAA desarrollada por Oller, es la arquitectura agente escogida para incluir el conocimiento que se tiene a priori sobre el cuerpo físico de un agente. Ella contempla tres tipos de capacidades en las que, dependiendo del nivel de abstracción de la información, se representan los módulos que forman al agente. Las que representan al cuerpo físico del agente, son las capacidades atómicas que incluyen conocimiento directo de la dinámica del cuerpo físico del agente. Estas capacidades, originariamente estaban definidas de manera que la información que contenían, no describía completamente el comportamiento del cuerpo del agente y en este trabajo, se ha procedido a redefinirlas aunque solamente para sistemas SISO.

En esta arquitectura se combinan las técnicas de la Inteligencia Artificial con la teoría de control tradicional, necesidad establecida por Zhang y Mackworth en [13] para desarrollar sistemas físicos inteligentes.

Las ventajas de incluir la dinámica del cuerpo físico del agente en las capacidades atómicas respecto de los trabajos existentes en la actualidad, son:

- Respecto de las CN desarrolladas por Zhang y Mackworth [13] es que incluir el conocimiento dinámico en la decisión del agente parece asequible y se podría tener un ejemplo funcionando en poco tiempo.
- Respecto de la arquitectura *subsumption* desarrollada por Brooks es que el comportamiento global del sistema se consigue con la evolución del sistema multi-agente, ya que los agentes interaccionan entre ellos para alcanzar las metas colectivas e individuales. Además aquí ya no hay comportamientos que luchan entre sí para tomar el control del agente.

Las desventajas de esta aproximación en la actualidad, residen en:

- La definición de los atributos que forman las capacidades, por dos razones, la primera comprobar fehacientemente la generalidad de éstos para sistemas SISO y segundo, la ampliación a sistemas MIMO.

- La implementación del algoritmo de decisión, sobre todo respecto al que hacer cuando no exista una capacidad atómica que refleje los requerimientos de una acción; y el cálculo del coeficiente de certeza que acompaña a las capacidades.

Finalmente se cree que esta representación permitirá, en cierta medida, definir una interfaz entre dos grandes ámbitos como son la Inteligencia Artificial y la Teoría de Control; y a partir de ella, el tratamiento eficientemente de problemas relacionados con sistemas complejos con restricciones temporales situados en entornos dinámicos y físicos.

8 Propuesta de Tesis.

El objetivo principal de la tesis es el de proporcionar al agente del conocimiento necesario para que, sabiendo a priori las características de la dinámica de su cuerpo físico, sea capaz de tomar decisiones físicamente realizables. Así, cada vez que se comprometa con sus semejantes, tendrá una certeza sobre la acción a la que se compromete basada en las restricciones inherentes a su cuerpo físico.

La idea reside en que el ingeniero de sistemas especifique al ingeniero de control, los requerimientos que necesita que su sistema cumpla, y éste último, los transforme en las especificaciones de control y diseñe los controladores adecuados para cumplir con éstos. Después, complete una ficha técnica de cada controlador y que éstas se conviertan en capacidades atómicas del agente.

Basado en estas capacidades atómicas, el ingeniero de sistemas dotará al agente de una cierta inteligencia que le permita discernir si su cuerpo le permite conseguir el objetivo para el cual se ha comprometido.

Siguiendo con esta idea, los objetivos parciales a cumplir durante el desarrollo de la tesis son:

1. Definir una manera simple de ejecutar las diferentes capacidades atómicas.

Cada capacidad atómica está relacionada con un controlador, por lo tanto cuando un agente se compromete a realizar una acción, teniendo en cuenta sus capacidades, deberá elegir el controlador que más se adecue para conseguir la respuesta dinámica esperada. El cambio de controladores, si no se realiza de manera específica, puede producir transitorios no deseados. La idea es determinar una manera sencilla de conseguir que el agente cambie de controladores.

2. Extender el análisis a sistemas MIMO.

Como la idea es poder representar la dinámica de cualquier sistema físico, es necesario extender el estudio del presente trabajo, a sistemas con más de una entrada, una salida, y modificar las capacidades atómicas, si es necesario.

3. Determinar una metodología para utilizar el conocimiento contenido en las capacidades atómicas en el razonamiento del agente físico.

Una vez representada la dinámica del cuerpo físico del agente en las capacidades atómicas, hay que buscar la manera de emplear este conocimiento. Una primera aproximación, sería utilizar razonamiento basado en casos (Cased-Based Reasoning) para escoger la capacidad que más se adecue a las especificaciones requeridas por el agente para ejecutar la acción.

4. Aplicar la arquitectura agente a la plataforma de robots futbolistas.

Teniendo las capacidades atómicas, el método de ejecución de las capacidades y el modo de usarlas, aplicar la arquitectura agente a un sistema real, concretamente la plataforma de robots futbolistas, conjunto de robots móviles y autónomos, cuya misión es la de emular un partido de fútbol.

9 Bibliografía.

- [1] Asada, M., Kuniyoshi Y., et al. “*The RoboCup Physical Agent Challenge*”. First RoboCup Workshop in the XV IJCAI-97 International Joint Conference on Artificial Intelligence. Pág. 51-56, 1997.
- [2] Brooks, R. “*Intelligence without reason*”. IJCAI’91. Pág. 569-595. 1991
- [3] Brooks, R. A., “*New Approaches to Robotics*”, Science, Vol. 253. Pág.1227-1232. September 1991
- [4] Brooks, R. “*Intelligence without representation*”. Artificial Intelligence 47. Pág. 139-159. 1991.
- [5] De la Rosa J. Ll., García R., Innocenti B., et al. “*Rogi Team Real: Research on Physical Agents*”. 3rd Workshop on RoboCup. 16th IJCAI. 1999.
- [6] De la Rosa, J., Innocenti, B., et al. “*An Example of Dynamical Physical Agents*”. RoboCup Workshop, European Championship Amsterdam 2000. 2000.
- [7] Jennings, N. R., Sycara, K., Wooldridge, M. “*A Roadmap of Agent Research and Development*”. Autonomous Agents and Multi-Agents Systems. Pág. 7-38. Editorial Kluwer Academic Publishers. 1998.
- [8] Moulin, B. Chaib-Draa, B. “*An Overview of Distributed Artificial Intelligence*”. Foundation of Distributed Artificial Intelligence. Editores G. M. P. O’Hare and N. R. Jennings. John Wiley & Sons, Inc. 1996
- [9] Oller A., de la Rosa J. Ll., del Acebo E., DPA: Architecture for Co-operative Dynamical Physical Agents, MAMA AW’99, June 1999.
- [10] Shoham, Y. “*Agent-oriented programming*”. Artificial Intelligence 60. Pág. 51-92. 1993.

- [11] Wooldridge, M. Jennings, M. “*Intelligent Agents: Theory and Practice*”. Knowledge Engineering Review. 1994.
- [12] Wooldridge, M. “*Intelligent Agents*”. Multiagent Systems. Editor G. Weiss, The MIT Press, April 1999.
- [13] Zhang, Y. Mackworth, A. “*Will the robots do the right thing?*”. Technical Report TR 92-10, UBC. 1992.
- [14] Zhang Y., Mackworth A.K. “*Specification and verification of constraint-based dynamic systems*”. Principles and Practice of Constraint Programming, Lecture Notes in Computer Science N° 874. Pág. 229-242, 1994.
- [15] Zhang Y., Mackworth A.K. “*Constraint Nets: A Semantic Model for Hybrid Dynamic Systems*”. Theoretical Computer Science 130. Pág. 211-239, 1995.
- [16] Zhang Y., Mackworth A.K. “*A Multi-level Constraint-based Controller for the Dynamo98 Robot Soccer Team*”. Proceedings of the 2nd RoboCup Workshop. Pág. 353-359, 1998.
- [17] Dorf, R., Bishop, R. “*Modern Control Systems 7th Edition*”. Addison-Wesley Publishing Company. 1995.
- [18] Kuo, B. “*Sistemas de Control Automático 7^a Edición*”. Editorial Prentice Hall. 1996.
- [19] Ogata, K. “*Ingeniería de Control Moderna 3^a Edición*”. Editorial Prentice Hall. 1998.
- [20] Phillips Ch., Harbor, R. “*Feedback Control Systems 2nd Edition*”. Editorial Prentice Hall Internacional. 1991.
- [21] Shinskey, F. “*Process Control Systems 3rd Edition. Application, Design and Tuning*”. Editorial McGraw-Hill Publishing Company. 1988.

10 Bibliografía Complementaria

- [22] Asada M., Kuniyoshi Y., et al. “*The RoboCup Physical Agent Challenge*”. First RoboCup Workshop in the XV IJCAI-97 International Joint Conference on Artificial Intelligence. Págs.51-56, 1997.
- [23] Barman R. A., Kingdon S. J et al. “*Dynamite: A Testbed for Multiple Mobile Robots*”. Proceedings of the IJCAI-93 Workshop on Dynamically Interacting Robots. 1993.
- [24] Brooks, R. “*Artificial Life and Real Robots*”. “Toward a Practice of Autonomous Systems: Proceedings of the First European Conference on Artificial Life” F. J. Varela and P. Bourguine, eds. MIT Press, Cambridge, MA. Pág 3-10. 1992.
- [25] Brooks, R., Stein, L. “*Building Brains for Bodies*”. Autonomous Robots 1. Págs. 7-25. 1994.
- [26] Brooks, R. “*A Robust Layered Control System for a Mobile Robot*”. IEEE. J Rob Autom. 2. Págs. 14-23. 1986.
- [27] Brooks, R. “*How To Build Complete Creatures Rather Than Isolated Cognitive Simulators*”. Architectures for Intelligence, K. VanLehn (ed)}, Erlbaum, Hillsdale, NJ, Págs. 225--239. 1989.
- [28] Brooks, R. “*Challenges for Complete Creature Architectures*”. First International Conference on Simulation of Adaptive Behavior, Págs. 434-443. 1990.
- [29] Brooks, R. “*Elephants Don't Play Chess*”. Robotics and Autonomous Systems Vol. 6, Págs. 315, 1990.
- [30] De la Rosa J. Ll. “*Heuristics for Cooperation of Expert Systems, Application to Process Control*”. PhD Thesis. Universitat Autònoma de Barcelona (UAB), 1993.

- [31] De la Rosa J. Ll., Oller A., et al. “*Soccer Team based on Agent-Oriented Programming*”. Robotics and Autonomous Systems. Ed. Elsevier. Vol 21. Pág. 167-176. Octubre 1997.
- [32] Inoue, K et al. “*Iterative Transportation by Cooperative Mobile Robots in Unknown Environment*”. Intelligent Autonomous Systems. Págs 30-37. 1998.
- [33] Johnson J., de la Rosa J.Ll., and Kim J.H., “*Benchmark Tests in the Science of Robot Football*” Proceedings IEEE of Mirost-98. Pág. 85-89. 1998. R. J. Stonier (ed). Univ. Central Queensland.
- [34] Johnson J., de la Rosa J.Ll., and Kim J.H., “*Benchmark Tests of Robot Soccer Ball Control Skills*” Proceedings IEEE of Mirost-98. Pág. 91-93. 1998. R. J. Stonier (ed). Univ. Central Queensland.
- [35] Kaelbling L.P., Littman L.M and Moore A.W. “*Reinforcement Learning: A Survey*”. Journal of Artificial Intelligence Research. Pág. 237-285. Mayo 1996.
- [36] Kitano H., Veloso M., et al. “*The RoboCup Synthetic Agent Challenge 97*”. XV IJCAI-97 International Joint Conference on Artificial Intelligence. Vol 1. Pág. 24-29. Agosto 1997.
- [37] [Kube and Zhang 1992] Kube C.R, Zhang H., Collective Robotic Intelligence, Second International Conference on Simulation of Adaptive Behaviour, pp 460-468, 1992 .
- [38] Mackworth, A. “*On Seeing Robots*”. “Computer Vision: Systems, Theory and Application”. Basu, A. and Li, X. Eds. Singapore: World Scientific Press. Pág. 1-13. 1993.
- [39] Mackworth, A. “*The Dynamics of Intelligence: Constraint-Satisfying Hybrid Systems for Perceptual Agents*”. AAAI Spring Symposium in Hybrid Systems and AI. 1999.
- [40] Mataric M.J. “*Interaction and Intelligent Behaviour*”. PhD dissertation. MIT. Mayo 1994.

- [41] Nagendra, M., Prasad,V., Lesser,R. and Lander,S. “*Learning Experiments in a Heterogeneous Multi-agent System*”. IJCAI-95 Workshop on Adaptation and Learning in Multiagent Systems. 1995.
- [42] Nagendra, M., Prasad, V. Lesser,R. and Lander, S. “*Learning organizational roles in a heterogeneous multi-agent system*”. In *Adaptation, Coevolution and Learning in Multiagent Systems: Papers from the 1996 AAAI Spring Symposium*. Pág. 72-77. Marzo 1996. AAAI Press.
- [43] Pagello, E. Et al. “*Emergent Cooperative Behavior for Multirobot Systems*”. *Intelligent Autonomous Systems*. Págs 42-52. 1998.
- [44] Steels L. “*Discovering the competitors*”. *Journal of Adaptive Behaviour* 4(2). 1996.
- [45] Unsal C. “*Intelligent Navigation of Autonomous Vehicles in an Automated Highway System: Learning Methods and Interacting Vehicles Approach*”. PhD Thesis, Virginia Polytechnic Institute. 1997.
- [46] Vlacic,L. et al. “*Intelligent Autonomous Systems: Griffith University’s Creation*”. *Intelligent Autonomous Systems*. Págs 53-60. 1998.