Examples of Dynamical Physical Agents for Multi Robot Control

B. Innocenti, J.Ll. de la Rosa, J. Vehí, I. Muñoz, M.Montaner, M. Fàbregas, A. Figueras, J.A. Ramón

Institut d'Informàtica i Aplicacions
Universitat de Girona
Av. Lluís Santaló s/n
10071 Girona - Spain
fax : +34 – 972 41 80 98 e-mail :{bianca,peplluis,vehi,imunoz,mmontane,marti,figueras,jar}@eia.udg.es

Keywords: Multi-Agent Systems, Physical Agents, and Multi Robot Control.

Abstract

Lastest trends on Artificial Intelligence (AI) lead to combine AI with the traditional Control Theory to obtain intelligent systems. The goal of this work is to find some parameters that describe dynamics of the physical body of any agent, and to use them in a decision algorithm to let the agent know about its physical limitations. As a first approach, dynamics are described for single input-single output (SISO) systems. The parameters should be generics, comparable and understandable to both, the agent (computationally treatable) and the human being.

1 Introduction

Nowadays some Artificial Intelligence (AI) techniques are being applied to control complex systems. Since Brooks [brooks1] and Zhang [zhang1] stated that the intelligence depends on the interactions with the environment, several researchers have been trying to combine AI tools with traditional Control Theory with the aim of developing intelligent robots.

New tendencies lead to control complex systems using agents and to consider the whole process as a multi-agent system that needs co-ordination and co-operation to obtain the desired results.

One language that allows programming agents is Shoham's AGENT0 [shoham]. In this language the state of an agent consists of components such as *capabilities* (things that the agent can do), *beliefs* (beliefs of the world, itself or other agent), *commitments* (commitments with other agents or itself) and commitment rules (settle how the agent acts). A commitment rule can be as the following:

```
COMMIT(
   (agent,REQUEST,DO(action, time)) ──► message condition
   (B,[now,Friend agent] AND          ┐
      CAN(self, action) AND           ├ mental condition
      NOT[time,CMT(self, no_action)] ),┘
      self, DO (time, action))
```

This rule can be read as:

If I receive a message from ***agent*** *which* ***requests*** *me to* ***do action*** *at* ***time*** *and I believe (**B**) that:*

- ***agent*** *is currently a* ***friend***
- *I* ***can do*** *the* ***action***
- *at* ***time*** *I'm not committed (**not cmt**) to do any other action,*

then ***commit*** *to* ***do action*** *at* ***time****.*

With the aim of achieving its commitments, an agent must check whether they are feasible or not. So before committing it compares the required action with its beliefs and capabilities.

When an agent has a physical body, not only does the performance of an action depend on the dynamics of this physical body but also that what is heuristically possible to do may result in non-desired consequences.

Recalling the capabilities represent the actions that the agent can do, they seem the appropriated '*mental state*' to represent the dynamics of the physical body.

The goals of this work are to find some kind of co-ordination between the AI techniques and the Control Theory and to analyse the behaviour of the whole system. The proposal here is to include some features of the dynamics of the agent physical body into the decision algorithm to get secure and reachable commitments.

This paper is organised as follows. Section 2 explains what physical agents are. Section 3 resumes the relevant aspects of the agent architecture used in this work. Atomic capabilities attributes are defined in section 4. Section 5 presents an example of fulfilling atomic capabilities. And section 6 concludes.

2 Physical Agents

According to Asada [asada], the meaning of having a physical body can be summarised as follows:

- Sensing and acting capabilities are not separable, but tightly coupled.
- In order to accomplish a given task, the sensor and actuator spaces should be abstracted under resource-bounded conditions (memory, processing power, controller, etc.).
- These abstractions depend on the interactions of the agent with the environment.

- The consequence of the abstraction is agent-based subjective representation of the environment.
- In the real world, both inter-agent and agent-environment interactions are asynchronous, parallel and arbitrary complex.
- Natural complexity of physical interactions automatically generates reliable sample distribution of input data for learning.

Based on these statements, researchers have developed several architectures to control robots. As examples, it can be mentioned among others the Brooks' Subsumption Architecture [brooks2] and Zhang and Mackworth's Constraint Net (CN) [zhang2]. In the former the reactive agents have a layered set of different behaviours that compete to take the robot control. In the later the robot, its controller and the environment are modelled as three different machines with input and output modules; based on this CN and the properties required for the controller, specified as a set of constraints, it's possible to automatically generate a controller with the desired specifications.

There are also several hybrid architectures that include reactive and deliberative behaviours as the Oller's Dynamical Physical Agent Architecture (DPAA) [oller].

3 Dynamical Physical Agent Architecture

The DPAA has been developed for physical agents and has several requirements that are built-in and that enables the agents to work in a real world, in real-time. Some of them are:

- ***Situated behaviour:*** agents must recognise asynchronous events and react both on time and in a proper way taking into account its physical body.
- ***Goal-oriented behaviour:*** agents must choose actions based on the whole system objectives and on its own.
- ***Efficiency:*** tasks must be executed efficiently considering the real physical odds that agents have to achieve them.
- ***Co-ordination:*** agents must keep in mind the positive and negative interactions with other agents.

To deal with all of these requirements, it is proposed the architecture shown in Figure 1.

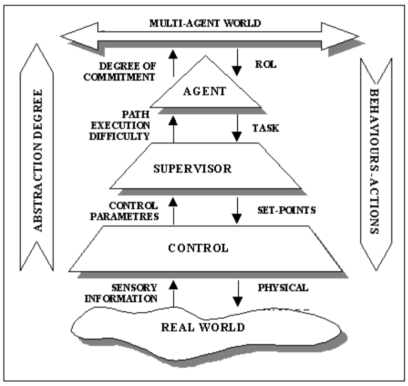

Figure 1: DPAA Architecture.

The DPAA is a layered architecture formed by three specialised modules:

- ***Control module:*** it's the direct connection of the agent with the real world.
- ***Supervisor module:*** it's the interface among the parameters of the agent real world with the agent logic world.
- ***Agent module:*** it's the connection with the multi-agent world.

It can be seen in the figure the different layers of the architecture, the increase of information abstraction degree as layers became logical and the decrease of the behaviour-actions as layers go to the real world.

After negotiating with other agents and in order to make a decision, an agent must check some external and internal parameters. The agent can get the external ones from the other agents by exchanging information. On the other hand, the internal ones must describe the different states of the agent physical body, both in low and high levels.

In order to include these internal parameters in the agent capabilities, three different kinds of capabilities depending on the abstraction level of the information are proposed, which are:

- ***Atomic capability:*** it contains information about the agent physical body, the perception of the environment through its body and the agent adaptation to the environment (learning).
- ***Basic capability:*** formed by several atomic capabilities, represents the knowledge of the supervisor. Information starts to be symbolic.
- ***Symbolic capability:*** it contains an abstract model of the world. Information is symbolic and depends on the agents' application.

Before deciding, an agent looks up in its set of capabilities and if they have enough information about the pretended action, the agent accepts or rejects the proposal. In case that the information is not complete, the agent communicates with the immediate lower level, and so on, to get it. In this way, when the agent accepts the action, it is aware of what its physical body can do with a high level of certainty.

In this approach, the parameters that formed the atomic capabilities turn out to be incomplete, so the contribution of this work is to complete them.

4 Atomic Capabilities

When trying to complete the atomic capabilities, two questions arise: What kind of information should the capabilities contain? Which parameters computationally treatable by the agent are the ones that best represent the dynamics of its physical body?

In order to obtain the characteristics of the dynamics of a physical agent, it has been necessary to do a complete study of the responses of different controlled systems. Because of the complexity of studying real systems in Control Theory [dorf]

[kuo] [ogata] [phillips] [shinskey], the scope of the present work has been limited to SISO systems.

In Control Theory before designing the controller for a system, Control Engineers should know the specifications that the response of the system must achieve. These specifications describe the response of the controlled system, so they can be used to complete the atomic capabilities. But this information has been modified in order to accomplish some requirements such as:

- Knowledge contained in the capabilities must be general. That means capabilities can be completed for any controlled SISO system.
- Atomic capabilities must be comparable between them. This implies that the parameters must be independent from the input, kind of system and between them. They must also ensure that the comparison is suitable.
- Information must allow computationally treatment to be understandable by the agent.
- Capabilities must be simple in order to be understood by control and system engineers.

Having these requirements the proposal is to complete the atomic capabilities with attributes that contains information about the temporal and frequency response of the controlled system, about the controller, about the open-loop system and about the actuators and the sensors. So, the capabilities are formed by the following attributes:

1. *Related to the controller:*
   - **Identification:** Controller name, as PID, predictive, etc.
   - **Controller type:** whether is linear or not.
   - **Controller structure:** Feedforward, multi-variable, control ratio, etc.

2. *Related to the open-loop system:*
   - **Order and type:** number of poles of the open-loop system and number of poles at the origin.
   - **Delay:** approximate time that goes by since a different input signal is applied until a change on the output of the system is observed.
   - **Gain:** deviation of the output value in steady state respect to the input signal.
   - **Time constant:** Time that takes a first-order system to get the 63% of the output value. It indicates how fast the system temporal response is.

3. *Related to actuators and sensors:*
   - **Sort:** kind of actuator or sensor (mechanic, electric, chemical, etc.)
   - **Precision:** interval in which the given magnitude can be erroneous.
   - **Sensibility:** minimal variation of the input that can be detected by the sensor or to which responds the actuator.
   - **Time constant:** time that indicates how fast answers the actuator or sensor to changes on the input signal.
   - **Hysteresis:** deviation of the magnitude value depending on whether it is reached by an increasing or a decreasing continuous change of the input.
   - **Temperature dependence:** change on the output value due to a different temperature from nominal.
   - **Linearity interval:** interval in which the actuator or the sensor works on its linear zone.
   - **Delay:** delay between a change on the input and its corresponding effect on the output.
   - **Noise rejection:** maximal power of the noise signal that does not affect the sensor or actuator output signal.

These attributes are included in atomic capabilities to be used in future applications but not in the scope of this work. The following attributes have been modified, redefined or adapted to reach the requirements above mentioned and will be immediately used in the decision algorithm:

4. *Related to the temporal and frequency response of the closed-loop system:*
   - **Precision**
   - **Overshoot**
   - **Rapidity**
   - **Persistence**
   - **Robustness**
   - **Aggressiveness**
   - **Control effort**
   - **Coherence**
   - **Identification**

4.1 Precision

This attribute is related to the deviation that the controlled system has respect to a ramp input signal with a τ slope at 2 τ times, being τ the time constant of the open-loop system.

$$Precision = 100 - \lim_{t \to 2\tau} \frac{\tau * t - y(t)}{\tau * t} * 100$$

This parameter has been defined in this way because it is possible to avoid infinite or zero values (the error is calculated for a time equal to twice the open-loop time constant). And also, it is independent of the kind of input applied (therefore can be compared with its equals).

4.2 Overshoot

As in Control Theory this attribute represents the relative value of the maximal value of the output signal respect to the steady state value. It is calculated as follows:

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

Where

t_p is the time at which the maximum value of the temporal response is produced.

If the temporal response do not present an overshoot then this parameter is 0%.

4.3 Rapidity

This attribute is a ratio between the time needed by the controlled system to get the steady state when there is a change on the set point and the same time but in open-loop. It is defined as:

$$Rapidity = \frac{t_{slc}}{t_{sla}}$$

Where

t_{slc}: closed-loop system settling time.

t_{sla}: open-loop system settling time.

Settling time: it's the time that requires the system to maintain the output between an interval of 2% or 5% of the steady state value.

The lower this value is the faster the systems responses.

4.4 Persistence

This attribute is related to the capability of the system to reject disturbances, which is to maintain the output signal within an acceptable value.

It has to be said that the disturbance rejection is sometimes a specification for designing the controller, so its evaluation will depend on the Control Engineer judgement. Anyway, a formula to calculate is provided for the two most common disturbances, which are step and pulse types.

In the case of step perturbations of amplitude A, the following way to calculate the persistence is proposed:

$$Disturbance = \left[\left(1 - \frac{IAE}{A \times \tau}\right) \times percentage\_disturbances\right]\%$$

Where

$IAE = \int_{t_1}^{t_2} |e(t)| dt$ It's the integral of the absolute error value.

A is the amplitude of the step.

τ the open-loop time constant.

The choice of τ is because it does not change as the closed-loop time constant does (depends on the controller and hence this attribute won't be independent).

To calculate persistence for a pulse disturbance:

$$Disturbance = \left[\left(1 - \frac{IAE}{B}\right) \times percentage\_disturbances\right]\%$$

Where

IAE it's the integral of the absolute error value.

B pulse area: pulse amplitude x pulse duration.

In both cases if the equation between parenthesis is negative, the persistence takes 0% value. That is that the system does not reject disturbances.

In case that there exists more than one kind of disturbance, this index will be the maximum value of all of them.

4.5 Robustness

This attribute represents the capability of the controlled system to maintain the output within acceptable values when there are variations in the parameters of the open-loop system or non-modelling dynamics.

The phase and gain margins give a magnitude of the system stability. They provide the maximum change that can have the parameters of the open-loop system to maintain stable the closed-loop system.

To calculate robustness it is necessary to know the phase and gain margins (PM_{nom} and GM_{nom}) of the system without variations on the open-loop parameters and both margins (PM and GM) with the maximum variations of the parameters. So the formula is:

$$Robustness = \frac{\frac{MP}{MP_{nom}} + \frac{MG}{MG_{nom}}}{2}$$

4.6 Aggressiveness

This attribute represents the system speed to respond to changes in the set point. It is defined as the percentage relation between the rising time (t_r) and the settling time (t_s) of the closed-loop.

$$Agressiveness = 100\% - \frac{t_r}{t_s} * 100\%$$

4.7 Control effort

This attribute describes the effort that the controller needs to keep the output in the desired value. Its evaluation is made as:

$$Control\_effort = \frac{IADU}{u_{max} - u_{min}}$$

Where

$IADU = \int_{t1}^{t2} \left| \frac{du(t)}{dt} \right| dt$ it's the integral of the absolute value of the derivative of the control signal.

u_{max} is the maximum value that can take the control signal.

u_{min} is the minimum value that can take the control signal.

4.8 Coherence

This attribute is related to the work interval in which the designed controller satisfies the required specifications.

$$Coherence = work\_interval$$

4.9 Identification

This attribute is added to identify the controllers that deal with the same inputs-outputs units in order to compare only the capabilities of the same sort of controller. That is, if an agent has several position and speed controllers, and the currently task needs a speed controller, the comparison among capabilities should be done only for that ones that represents speed controllers. It is defined as:

$$Id = input\_units, output\_units$$

5 Example of Capability

Assuming the system open-loop transfer function as:

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

controlled by a PID with the approximate derivative with the following constants:

Kp = 150	Ki = 40
Kd = 50	N = 50

perturbed 95% of times by a pulse of amplitude 10 and duration of 20 sec., and with a non-modelled pole in

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

let's complete the atomic capability associated to this controller. The simulated response of the open-loop system to a step set point of amplitude 3 is:

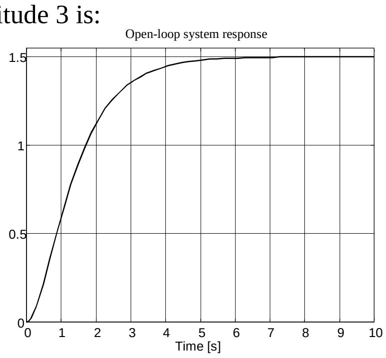

Figure 2: Open-loop system temporal response.

The open-loop time constant of the system is:

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

and the open-loop settling time is:

$$t_{sla} = 4.6s$$

The controlled system response is depicted in Figure 3, with the pulse disturbance affecting it.

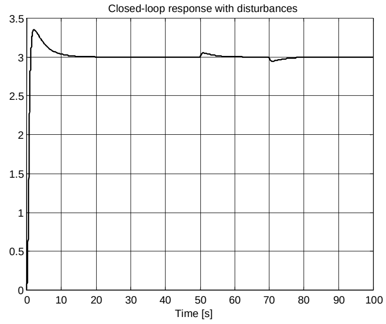

Figure 3: Controlled system response with disturbances.

From this simulated response, it is possible to calculate

The overshoot as

$$M_p = \frac{3.36 - 3}{3} \times 100\% = 12\%$$

The closed-loop settling time

$$t_{slc} = 8.6s$$

The rise time

$$t_r = 0.8945s$$

And the IAE produced by the pulse disturbance:

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

The closed-loop response to a ramp input of τ slope is shown in Figure 4.

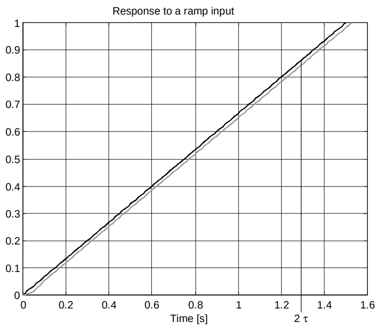

Figure 4: Closed-loop system response to a ramp input.

Considering this response, the system error in 2 τ is:

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

Apart from these figures, and applying some formulae cited in this paper is possible to reckon:

The area of the pulse disturbance:

$$B = amplitude \times duration = 10 \times 20 = 200s$$

The IADU:

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

The nominal phase and gain margins:

$$MF_{nom} = 51.0080^{o}$$

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

And the same margins but considering the non-modelled pole:

$$MF = 46.3911^{o}$$

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

With these values and making use of the different formulae of the attributes let's complete the atomic capability of this controller:

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

$$Overshoot = Mp = 12\%$$

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

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

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

$$Agressiveness = 89.5988\%$$

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

$$Coherence = [0,6]$$

$$Id = cm/sec, cm/sec$$

The atomic capability of this controller is completed with these values and with the corresponding ones of the open-loop system, sensor and actuator and controller.

6 Conclusion

In this paper, one way to include knowledge about the physical body into the states of an agent is presented. Thus the agent has enough information about its dynamics to decide feasible actions.

The idea is to have a set of controllers installed on the agent's body (and a set of atomic capabilities associated with them) that modifies its dynamics in a desired way. So, before committing itself to an action, the agent inspects its own capabilities and according to its physical constraints makes a decision.

Any commitment includes more than only its physical body constraints; it has to consider the task the agent is doing, the current state of the environment and the modifications produced by agents on it. That's why, besides the atomic capabilities, the agent has basic capabilities and symbolic capabilities. And all of them must be included in a decision algorithm.

Next step is to include knowledge encompassed in the atomic capabilities into a decision algorithm and apply it to a real physical system to verify its applicability.

Currently this idea is being applied to the soccer benchmark proposed in the RoboCup initiative [asada] and in convoying of vehicles.

Results are available on:

http://eia.udg.es/~bianca/phyiscal_agents

Acknowledgements

This work is partially funded by projects TAP98-0955-C03-02 “Diseño de agentes físicos (DAFNE) / Physical Agents Design” and TAP99-1354-E, "Eurobot-Iberia: red temática en agentes físicos / Thematic Network on physical agents" of the Spanish Research Foundation CICYT, and "Laboratori de l'Equip Internacional de Robots Mòbils de Girona (RoGi), as special action 2000ACES00018 of the Catalan Government.

List of References

[asada] Asada, M. et al, “The RoboCup Physical Agent Challenge”, *First RoboCup Workshop in the XV IJCAI-97*, 51-56 (1997).

[brooks1] Brooks, R., “Intelligence without reason”, *IJCAI'91*, 569-595, (1991).

[brooks2] Brooks, R., “A robust layered control system for a mobile robot”, *IEEE Journal of Robotics and Automation* RA-2, 14-23 (April 1986).

[dorf] Dorf, R., Bishop, R. “Modern Control Systems 7th Edition”, Addison-Wesley Publishing Company, (1995).

[kuo] Kuo, B. “Sistemas de Control Automático 7ª Edición”, Prentice Hall Editors, (1996).

[ogata] Ogata, K. “Ingeniería de Control Moderna 3ª Edición”, Prentice Hall Editors, (1998).

[oller] Oller, A. et al., “DPA: Architecture for co-operative dynamical physical agents”, *MAMAAW'99*, (June 1999).

[phillips] Phillips Ch., Harbor, R. “Feedback Control Systems 2nd Edition”, Prentice Hall International, (1991).

[shinskey] Shinskey, F. “Process Control Systems 3rd Edition. Application, Design and Tuning”, McGraw-Hill Publishing Company, (1988).

[shoham] Shoham, Y., “Agent-oriented programming”, *Artificial Intelligence 60*, 51-92 (1993).

[zhang1] Zhang, Y, Mackworth, R., “Will the robots do the right thing?”, *Technical Report TR 92-10*, UBC, (1992).

[zhang2] Zhang, Y, Mackworth, R., “Constraint Nets: A semantic model for hybrid dynamic systems”, *Theoretical Computer Science 130*, 211-139, (1995).