Robotica (2001) volume 19, pp. 611–617. Printed in the United Kingdom © 2001 Cambridge University Press

Neural based RSPN multi-agent strategy for biped motion control

Amar Khoukhi

LS, 14 rue Amirat Abdelkader, Ouled-Moussa, Boumerdes (Algeria)
E-mail: khoukhia@hotmail.com

(Received in Final Form: January 8, 2001)

SUMMARY

In this paper fhe problem of motion control of a biped is considered. We develop a new method based on multi-agent associated Neural *AIGLS (On-line Augmented Integration of Gradient and Last Sguare method) – RSPN (Recursive Stochastic Petri Nets*) strategy. This method deals with organization and coordination aspects in an intelligent modeling of human motion. We propose a cooperative multi-agent model. Based on this model, we develop a control kernel named IMCOK (Intelligent Motion COntrol Kernel) which consists of a controller, a coordinator and an executor of different cycles of the motion of the biped. When walking, IMCOK receives messages and sends offers. A Decision Making of Actions (DMA) is developed at the supervisor level. The articulator agents partially planify the motion of the associated non-articulator agents. The system is hybrid and distributed functionally. The learning of the biped is performed using an *On-line Augmented Integration of Gradient and Last Sguare Neural Networks* based algorithm. In the conflictual situations of sending or receiving messages by the managers of MABS we apply a new strategy: *Recursive Stochastic Petri Nets (RSPN*). This module is fundamental in the On-line information processing between agents. It allows particularly the Recursive strategy concept. Cognitive agents communicate with reactive (non-articulator) agents in order to generate the motion.

KEYWORDS: Biped; Multi-Agent; Articulator and non-Articulator Agent; IMCOK; Neural Networks; On-line-AIGLS; RSPN; Artificial Life.

1. INTRODUCTION

Multi-agent systems (MAS) have an explicit representation of their environment on which they can reason and from which they can predict future events.[1] Many architectures and theories about cognitive and reactive agents have been proposed.[2–4] They are concerned with the behavior of a collection of autonomous agents aiming at solving a given problem.[4]

However, in the past, soft computing approaches have emerged[5] to solve problems for which the conventional computational approaches have been proven ineffective, i.e. in situations of failure of the classical mathematical approaches of modeling and then computing. For instance, if we try to control a biped with 25 degrees of freedom by using the conventional approaches based modeling, such as the Lagrange or the Newton-Euler formalisms, which give the relationship between inputs currents and Cartesian positions outputs of the biped, the classical deterministic or stochastic optimal or adaptive control methods will give us high strongly coupled and time varying non-linear differential equations for which stability is not guaranteed.[6] And even if we make some approximations on the model to solve the control problem, the time consuming and the lack of convergence of the control algorithms renders conventional approaches ineffective. So we have what we can define as the problem of *modelizability.*[6] Hence, there has been much interest in applying intelligent methods to solve these problems, especially the problem of motion control of walker robot.[6] For instance, in reference [7] the authors showed how to build a lifelike agent made of “neurons” to perform various tasks and survive in a simulated environment. Usually, a simple network cannot accommodate complex task performances.

Researchers have studied the problem of motion control of bipedal robots from many angles, such as autonomous locomotion,[8–10] by using methods-based phase of movement. Many researchers investigated biped and walking robots on a surface, like those that can climb a sloping surface.[10] Brooks[11,12] has developed the *Behavior Based Approach* implemented by using those flat *Augmented Finite State Machines* in order to design a *S-ARCHitecure* level between perception and action of the biped. This level is realized with a *Behavior Language*. But the implementation of the AFSM has many drawbacks[12] especially, lack of adaptability and on-line control possibilities. The second drawback is the possible explosion of the number of levels in the AFSM network. Pratt[13] has developed a new approach to control the biped motion based on exploiting the inherent robustness and natural dynamics. The use of multi-agents systems associated with neural approach has then shown the modeling is based on agents and the learning is based on artificial neural networks methodology. Hence, we consider principles and fundamentals of Distributed Artificial Intelligence (D.A.I.) in order to build a cooperation and communication system between agents. The biped learning is performed using the *On-line Augmented Integration of Gradient and Last Square algorithm*. To solve conflictual situations between agents of sending or receiving messages, we use the *Recursive Stochastic Petri Nets* strategy, a new method developed by the author.[14,15]

https://doi.org/10.1017/S0263574701003344 Published online by Cambridge University Press

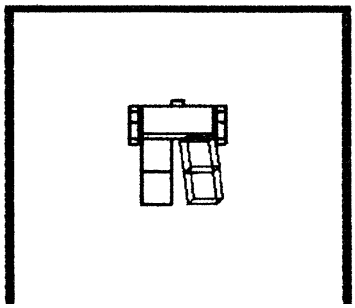


Fig. 1. Schematic representation of biped.

The model is called Multi-Agent Biped System (MABS) and the control kernel is called Intelligent Motion COntrol Kernel (IMCOK). The biped learns and recognizes different forms of movement. Actually, the coordination between the biped members causes its movement. Figure 1 is a representation scheme of a biped.[16]

2. SYSTEM DESCRIPTION

2.1. Definition

The MABS system is a hybrid and distributed in terms of function and scheduling, i.e. it consists of a set of hybrid agents (cognitive and reactive) that cooperate to accomplish functions of control of the biped. Agents use a dynamic organization strategy based on the use of the negotiation technique based on communication protocol DDE (Dynamic Data Exchange)-inter-agent through conversation with the server. This technique allows the exchange of messages between agents. Figure 2 shows the outline of MABS.

2.3. Operation levels

The scheme in Fig. 3 shows the different levels of our architecture. The user can communicate with MABS via an interfacing of communication which is the necessary means of communication. Figure 4 shows different actions of the MABS system.

2.4. Procedure of function

It is based on the following algorithm, followed by every agent:

- message receiving,
- scheduling,
- sending results.

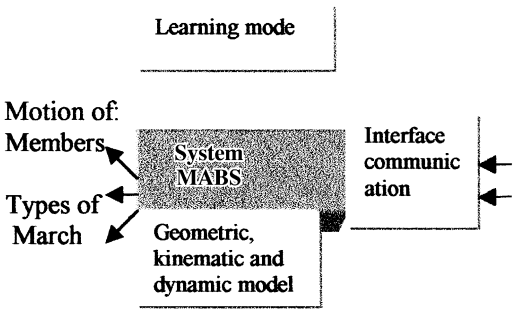


Fig. 2. Outline of MABS.

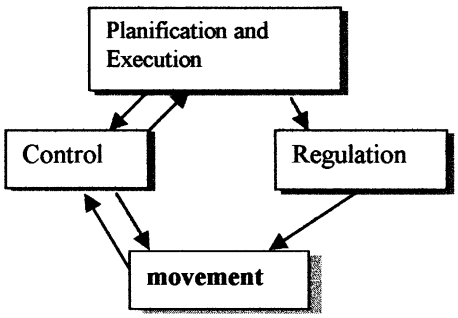


Fig. 3. MABS levels.

3. ARCHITECTURE OF THE MABS SYSTEM

MABS consists of (Fig. 5):

- Kernel (IMCOK) Intelligent Multi-agent COntrol Kernel that is composed of a supervisor agent, a PERCR agent, an equilibrator agent and a regulator agent.
- Mode of learning
- Tools of detection of a failure
- Interfacing of communication
- Articulator agents
- Non-articulator agents.

3.1. Articulator agents

These agents are homogeneous in their structures, but differentiate by their associated functions. Our articulator agent has three processes: communication, decision and scheduling (Fig. 6).

Process of communication: It receives the message coming from the supervisor or other components by the Process of Input Communication (PIC); it shapes it in the Internal Message File (IMF), and transmits the results of scheduling by the Process of Output Communication (POC).

Process of decision: at the receipt of a message from the regulator agent, the articulator agent decides after consulting the scheduling process, whether to send the control to the non-articulator agent or not.

Process of scheduling: this process consists of the library of plans: the necessary data allowing the process of decision either to send a signal of activation to the non-articulator agent or not, is contained in the library of plans (like max and min limits of rotation of the non-articulator agent partner). The ability of articulator agents resides in their plans. They do not have any process of decision; their primary function is clearly the execution of the movement, hence they represent reactive agents. They consist of:

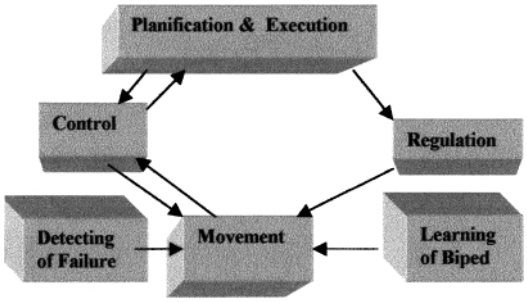


Fig. 4. The scheme of general actions.

https://doi.org/10.1017/S0263574701003344 Published online by Cambridge University Press

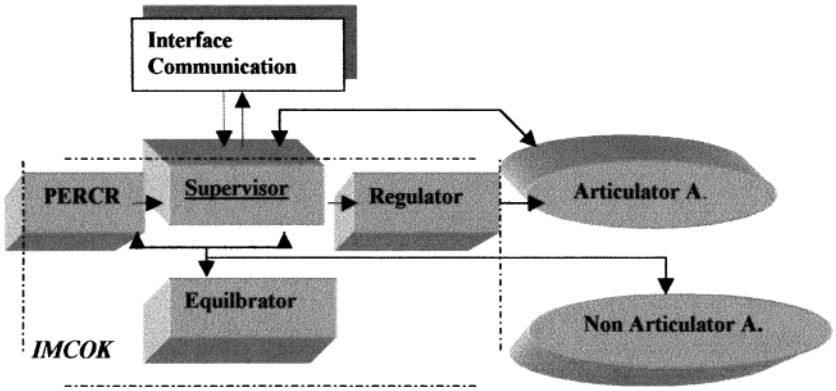


Fig. 5. MABS and IMKOC architectures of movement.

Process of communication: It consists of receiving signals of control from articulator agents in coordination with the supervisor in order to cause or not the action of motion.

Process of scheduling: The necessary data allowing the non-articulator agent to plan its trajectory of reference and then to follow it, is contained in the library of plans.

3.3. Regulator agent

It consists of a process of communication, a process of regulation, a motor process, and a process of scheduling.

Process of communication: It receives messages from the executor and sends answers to be processed by PERCR and the supervisor, then the regulator agent sends a message of starting or not the motion to the articulator agent.

Process of Regulation: its function is the regulation of speeds given by the supervisor.

Motor process: its unique function is the generation of a signal (message) called message motor.

Process scheduling: this process contains the necessary information for the treatment.

3.4. PERCR

After having received signals from the environment (reversed picture), the cerebellum of our system constructs a real picture of the environment. Then the resulting signal is transmitted to the supervisor.[7]

PERCR (PERCeption-Reception) receives two types of signals, one from the sensors, the other from the agents, and has two processes:

Process of perception: this process ensures the perception of the environment's signals.

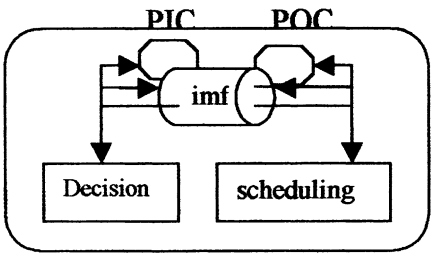


Fig. 6. Articulator agent .

Process of scheduling: it contains the library of plans: these are two essential data parts: data specified for the perception that ensures the management of the system of picture treatment, and data specified for control, which ensures the management of the control system by sending data concerning states of different members in progress during the movement.

3.5. Control Agent (IMCOK): (Intelligent MOtion COntrol Kernel)

This agent is responsible for the control of motion in the MABS system. In the initial state the supervisor sends a message to IMCOK (Action of Control) containing information on the states of the agents to be controlled. This information is: the address, the state, operation mode. . . . At each moment, IMCOK is carefully waiting for messages coming from articulator, regulator, and equilibrator agents, treat them, then generate a resulting signal to inform the user about the state of motion. If the motion is satisfied, the resulting signal is a success, and the supervisor prepares the next cycle of motion and determines members concerned by this cycle. Otherwise, the motion is not satisfied, and the signal is a failure. The supervisor will not then determine any member for the next cycle of motion and makes a signal to the user indicating failure, and IMCOK is reiterated until giving a satisfactory signal. When this is done, the proper signal is stored in the memory and reactivated automatically.

3.6. Supervisor

The supervisor receives the already quoted signals from agents, treats them and then generates new signals of command or answers (structures of solution to problems met by other agents of MABS). There are two types of communication at the level of the supervisor (Fig. 7):

The intrasupervisor communication (between its different constituents) effected via the Internal Messages File (IMF), and the inter-supervisor communication effected via DIspatcher of Signals (DISI), and Process of Output Communication (POC). Signals coming from different components are identified and recognized by the Receiver and Identifier of SIgnals (RISI). All components of MABS system are managed by their respective managers.

https://doi.org/10.1017/S0263574701003344 Published online by Cambridge University Press

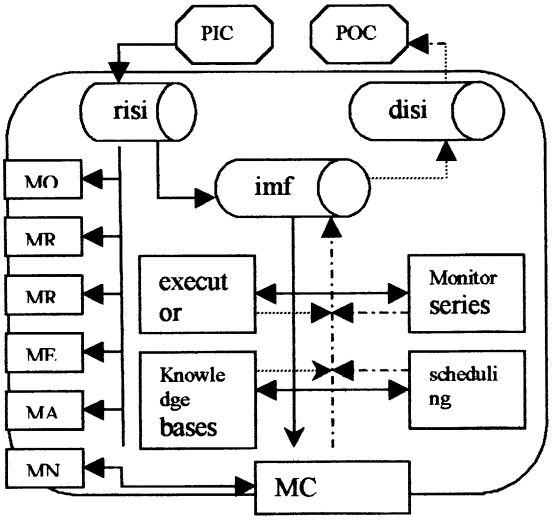


Fig. 7. Supervisor agent.

MC: manager of communication
ME: manager of equilibrator
MR: manager of regulation
MO: manager of control
MA: manager of articulator agent
MN: manager of non-articulator agent.

3.6.1. Basis of knowledge: It is the database of MABS. It contains all information on different components of the system (PERCR, regulator, equilibrator, articulator and non-articulator agents, and of each cycle of movement (4 cycles, two for double and two for single support).

3.6.2. Monitor of Series (MS): Its main role is the coordination between the intra-supervisor actions. We can associate different strategies to plan the activities of this monitor:

a. With static priority: once a priority is affected into a program, this one won't change until the end of the execution of this program.
b. With dynamic priority: the priority affected into program changes according to the environment of execution of this one (load of system, types of actions to be executed . . .).

3.6.3. Executor: The allocation of an executor to a program can be made according to priorities to program them automatically. For instance, an organ or the person responsible for the exploitation can give to every signal coming from another organ a priority; with its organization, the priority can be allocated statically or dynamically.

After execution, the result will be transmitted to the dispatcher of signals in order to transmit it to the concerned agent.

3.6.4. Receiver and Identifier of SIgnals (RISI): Contrary to other agents of MABS, the supervisor has a superior work load, which requires a Receiver and Identifier of SIgnals (RISI), a loaded element of identification, to send received messages from the environment into processes or the corresponding manager in the supervisor.

3.6.5. DIspatchor of SIgnals (DISI): Plays an important role in the diffusion of signals (either signals that have just been executed, or those that have just been managed without being executed).

3.6.6. Managers: They ensure the management of agents of the MABS system. The manager sends a message of working to the corresponding managed agent (Fig. 7). Failure to execute actions may occur. This corresponds to conflictual situations. In these cases, the managers solve conflicts by using the Recursive Stochastic Petri Nets Strategy; it allows one to find the optimal time to send the next action. Once the time and action are found, we repeat IMCOK.

3.7. Equilibrator

This process ensures the balance of the biped during the walk or when stationary. The physical conditions, so that the robot is in a state of balance, are that the center of gravity of the body is on the surface of the lift (surface taken between elements in contact with ground). It consists of a process of communication, process of planification and process of equilibration.

Process of communication: It is the same as that of an articulator agent; it ensures the communication with other agents

Process of planification: It contains a library of plans and an archiving memory.

The plans are to know if they contain all necessary information that allows the equilibrator to ensure its functions. Among these data are rotations, flexions and extension degrees intervals as a function of the center of mass of the biped, tools for the determination of the center of mass as a function of the situation of other organs. This is given by the response signal coming from articulator agents to the supervisor; relative data to the linking of intra-equilabrator actions. The archiving memory is used to store the results of equilibration.

Process of equilibration: Signals coming from articulator agents to the supervisor contain information about the degrees of flexion, rotation and extension of non-articulator agents. These data will be planified by the process of planification. For instance, compare at the level of plans the foot flexion to the interval of flexions permitting an equilibrium. If it is outside the interval of the center of mass, then it announces an unbalanced situation; it then sends a message to the manager of equilibration to treat this conflictual situation. The supervisor which contains the manager of equilibration will deal with conflictual situations by using the Recursive Stochastic Petri Nets based strategy, in order to find the next time and the next action to be performed.

4. COMMUNICATION

The MABS system, according to its architecture and its function, appears like a very necessary system in matter of communication needs. MABS agents communicate between them by sending messages.

https://doi.org/10.1017/S0263574701003344 Published online by Cambridge University Press

Every signal (message) represents information needed by an agent (a control to be achieved) and/or a result of the treatment of the previous agent (Fig. 8).

5. NEURAL APPROACH

Because of the high complexity of biped motion control, (we have 25 degrees of freedom), a large network size is needed. This may not be feasible by using classical Back-Propagation or Recursive Last Square algorithms, as they have slow convergence speed rates. The biped learns the form of movement by using the On-line AIGLS (Augmented Integration of Gradient and Last Square) control algorithm.[5,6] So MABS can be considered as having this form (Fig. 9) where Ds, Ts, and Ms are Degrees of freedom, Types of motion and Members of the biped, respectively, they are the inputs of our neural network.

All forms of movement learned by the biped are arranged in the supervisor agent and are the outputs of the neural network. At the supervisor level, the Neural AIGLS method[5,17] is applied for biped learning for several cases of motion.

$$\mathrm{Oj} = 1/\mathrm{l} + \exp(-\mathrm{netj});$$

$$\mathrm{netj} = \sum_i w_{ij} O_{ij};$$

$$\delta j = Sj(1 - Sj) \sum_K \delta_K wjk$$

$$Wij(n+1) = wij(n) + \alpha_{ijk} \delta jSi + P[(wij(n) - wij(n-1)]$$

$$\alpha_{ijk} = \alpha_{ijk-1} + \gamma(1 - jSi)$$

P is defined according to IGLS scheme,[5] and γ is a random number generated uniformly on [0,1].

6. RECURSIVE STOCHASTIC PETRI NET CONFLICTS SOLVING

For two decades, Information Processing Systems have evolved towards strongly massive parallelism, concurrence, synchronization, sharing and distribution resources and communication. Hence, qualitative and quantitative performances evaluation becomes imperative to design any production system, especially to deal with problems like blocking, breakdown of machines, etc., . . . A fundamental technique developed to treat these problems was the Stochastic Petri Nets (SPN), which are graphical models consisting of a set of resources (places) and a set of events or operations (transitions) such that to each event or transition is associated a random variable describing the duration of this transition. However, when the SPN to be analyzed is not bounded, the size of the model increases rapidly, giving what is known as a combinatory explosion. For instance, in our case study of bipedal robots and in managing the overall possible conflictual situations, we have 9 agents, each one with between three and four processes and each process has to be managed by eight managers. If an integer represents one octet, then the quantity corresponding to the whole computing will necessitate about 2500 GO of space in hard disc, which is enormous. Hence, we have associated the (RSPN) Recursive Stochastic Petri Nets[14,15] to study different situations of movements. In conflictual situations, we use this powerful RSPN strategy which allows an adaptability in the structure of the stochastic Petri net and permits one to avoid a combinatory explosion.[14,15] It has the following form:

After we have determined the associated graph, we determine all situation generating a conflict. So, we obtain (Fig. 10):[14,15]

P(Si + 1/Si) represents the probabilities of transition from state Si to state Si + 1.

P(Si) represents the probabilities that request is in states i

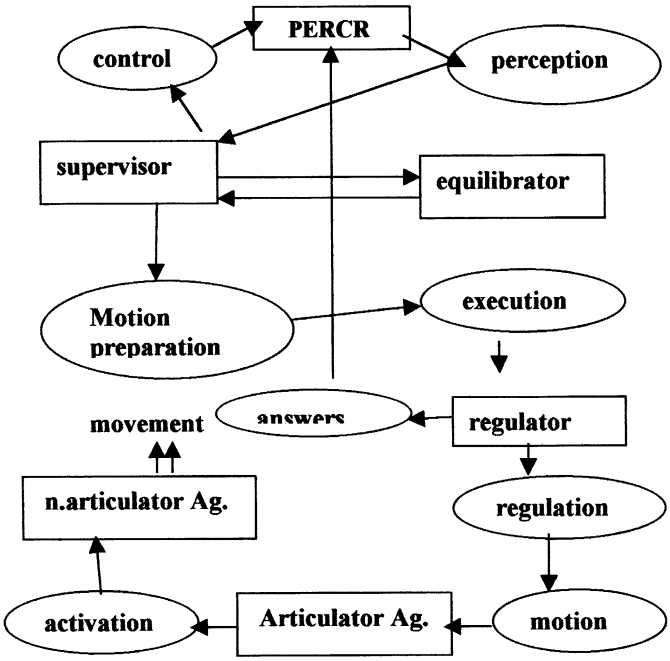


Fig. 8. Detailed scheme of MABS operation.

https://doi.org/10.1017/S0263574701003344 Published online by Cambridge University Press

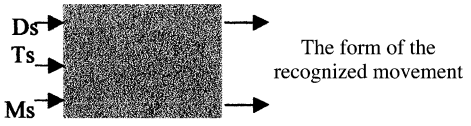


Fig. 9. The representation of neural biped system.

We have:

$$P.Q = 0, \sum_{1}^{n+2} pi = 1$$

then we can compute the average time in each place with:

$$T^*(Pi) = M^*(Pi)/Post(Pi,.).F^*$$

And the average tension of transformation[14,15] with:

$$F^*(ti) = \sum \lambda i(Mi)\pi i \quad j \in SMj$$

7. SIMULATION

Some results of simulation are explained re this scheme, initially in a state of walk. Inter-agents message exchanging protocol via DDE (Dynamic Data Exchange) through conversation with the server, thereby it transmits (transactions) to the server, the latter answers by providing data or services. We choose, then, asynchronic transactions. The general scheme of the algorithm is:

```
Client agent
{
   xtype_connect;
   put message in case
   transaction of emission
   read answer
   send xtyp_disconnect
}
```

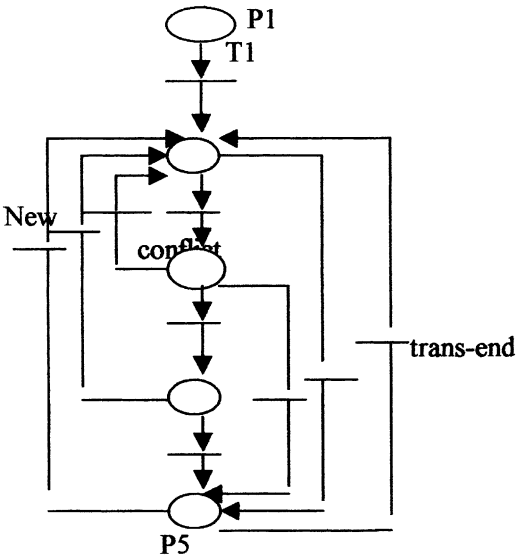


Fig. 10. Example of a Biped Conflict Solving Particular Situation for MABS.

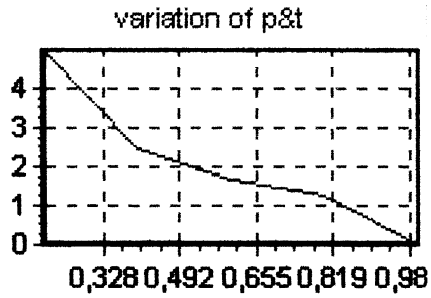


Fig. 11. Simulation results: Variation of the Conflict Solving Probability during the On-Line Biped Motion (at time zero the probability is 0.98 and at time 4.5 units the probability is 0.0).

```
server agent
{
xtyp_connect_confirm
xtyp_poke
read data from case
send answers
}
```

The schemes on Figs. 11 and 12 show some results:

Figure 11 shows the Variation of the Conflict Solving Probability during the On-Line Biped Motion (at time zero the probability is 0.98 and at time 4.5 units the probability is 0.0).

Figure 12 shows the outputs of the two following cases of study (Foot Left Flexion of 20°, Rotation of 8° of the biped about X axis).

8. CONCLUSION

In this work we have presented an original and new approach for the control of Biped Motion based on a Neural RSPN Multi-Agent Design. The initial model was named MABS Multi-Agent Biped System. MABS was produced after mixing four scientific fields of advanced research: walking bipedal robots,[17] multi-agent systems as a promising branch of distributed artificial intelligence,[18–20] advanced artificial neural networks methodologies and recursive stochastic Petri nets conflicts solving. Thus we have created all the auxiliary systems MABS, IMCOK, PERCR, On-line AIGLS, RSPN. . . . Thus, MABS is designed so that the following criteria are respected:

(i) Flexibility: the possibility of integrating new agents in the initial representation.

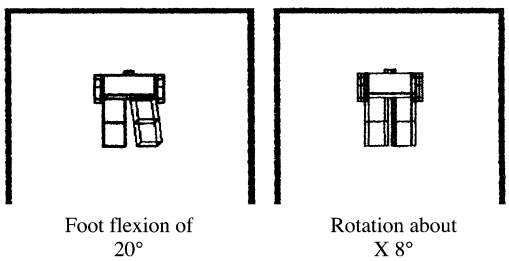


Fig. 12. Simulation results: outputs of the two following cases of study (Foot Left Flexion of 20°, Rotation of 8° of the biped about X axis).

https://doi.org/10.1017/S0263574701003344 Published online by Cambridge University Press

(ii) Programmability: agents are programmable tools, in fact; then can manipulate objects, data and methods.
(iii) Robustness and operation: which consists of the high level of related inter- and intra-agents and which has as a consequence resistance to many disturbances due to intrinsic or extrinsic environment of the Biped.
(iv) Task distribution: the distributed planning of the treatment through agents gives a best view of distributivity and executability tasks.

The presented work deals with collaboration and information exchanging between agents. Each agent is specialized in a precise task. Thus, following this approach the generation of motion is made by following a number of cycles of treatment beginning with a supervisor and finishing with non-articulator agents. This methodology, the author believes, is a step ahead in artificial intelligence.

References

1. J. Ferbery, "Reactive Distributed Artificial Intelligence: Principle and Application", In: *Foundations of Distributed Artificial Intelligence* (G.M.P. O'Hare and N.R. Jennings, Eds) (John Wiley & Sons, 1996).
2. B. Moulin and B. Caib-draa, "An Overview of Distributed Artificial Intelligence", In: *Foundations of Distributed Artificial Intelligence* (G.M.P. O'Hare and N.R. Jennings Eds) (John Wiley & Sons, 1996).
3. H. Van Dyke Parunak, "Applications of Distributed Artificial Intelligence in Industry", In: *Foundation of Distributed Artificial Intelligence* (G.M.P. O'Hare and N.R. Jennings, Eds) (John Wiley & Sons, 1996).
4. K. Fischer, J. Muller and M. Pischel, "A General Testbed Artificial Intelligence Applications", In: *Foundation of Distributed Artificial Infelligence* (G.M.P. O'Hare and N.R. Jennings, Eds) (John Wiley & Sons, 1996).
5. G.W. Ng, *Application of Neural Networks to Adaptive Control of Nonlinear Systems* (John Wiley & Sons, 1997).
6. A. Khoukhi, "*Systemes Neuro-Flous Motivation, Théorie et Applications*", (Ed, Bertie, Algeria, 2000).
7. A. Khoukhi, *et al*., "Multi-Agent Based Strategy for Biped Motion control", *ICRAM'99, Int'l. Conf. on Recent Advances in Mechatronics*, Istanbul Turkey (24–26 May, 1999) pp. 201–212.
8. S. Kagami, *et al*., "Design and Development of a Legged Robot Research Platform JROB-1", *IEEE Int'l Conf. On Robotics & Automation*, Leuven, Belgium (May, 1998) pp. 146–151.
9. J.S. Liu, L. Fu and J.Yang, "Multi-Agent Based Control Kernel for Flexible Automated Production System", *IEEE Conf. On Robotics & Automation*, Leaven, Belgium (May, 1998) pp. 2134–2139.
10. M. Rostami and Guy Bessonet, "Impactless Sagittal Gait of a Biped Robot During the Single Support Phase", *IEEE Int'l Conf. Robotics Automation*, Leuven, Belgium (May, 1998) pp. 1385–1391.
11. R.A. Brooks, "A Robot that Walks; Emergent Behaviors from Carefully Evolved Network", *Internal Report, No, 1-1-1091* (Artificial Intelligence Laboratory, MIT, February 1989).
12. R.A. Brooks and L A. Steini, "Building Brains Bodies", *Autonomous Robots*, **1**, No. 7, pp. 401–430 (1994).
13. J.E. Pratt, "Exploiting Inherent Robustness and Natural Dynamics in the Control of Bipedal Walking Robots", *PhD Dissertation* (Dept. of Elec. Eng. & Comp. Science MIT, USA, June 2000).
14. S. Donatelli, "Superposed Generalized Stochastic Petri Nets: Definition and Efficient Solution", *Proc. of the Int'l. Conf. on Theory and Application of Petri Nets*, Saragoza, Spain (June 20–24, 1994) pp. 258–277.
15. A.Khoukhi, "Recursive Generalized Stochastic Petri Nets, Theory and Applications", *Internal Report*, No. 107 S/98, (USTHB, Algiers, Algeria, 1998).
16. F. Noreils and S.R. Chatila, "Control of a Mobile Robot Actions", *IEEE Int'l Conference on Robotics & Automation*, Scottsdale, Arizona, (USA) (June, 1989) pp. 701–712.
17. C. Yu-Yung, L. Fu and C. Yu-Chien, "Multi-agent Based Dynamic Scheduling for a Flexible Assembly System", *IEEE Int'l Conf. on Robotics & Automation*, Leuven, Belgium (May, 1998) pp. 2122–2128.
18. D. Steiner, "An Integrated Environment for Constructing Distributed Artificial Intelligence Systems", *Foundation of Distributed Artificial Intelligence* (G.M.P. O'Hare and N.R. Jennings Eds) (John Wiley & Sons, 1996).
19. Y.F. Zheng and J. Shen, "Gait Synthesis for the SD-2 Biped Robot to Climb Sloping Surface", *IEEE Trans. on Robotics & Automation* (Feb, 1990), **6**, No. l, pp. 86–96.
20. A. Khoukhi and L. Khoukhi, "Fuzzy Multi-Agent based Strategy for the Control of a Biped Motion", *MHS '99, Int'l Conference on Micro-Mechatronics and Human Sciences*, Nagoya, Japan (Nov., 1999) pp. 22–25.

https://doi.org/10.1017/S0263574701003344 Published online by Cambridge University Press