Networked multi-agent approach for redundant manipulator control

Vuk Vujović, *School of Electrical Engineering, University of Belgrade*
Kosta Jovanović, *School of Electrical Engineering, University of Belgrade*

***Abstract*—This paper investigates distributed approach to control of redundant robotic manipulator. As opposed to usual control that is based on knowing the whole model of the kinematic chain, idea of distributing the error information to local controllers is used. With multi agent approach even though no individual robotic joint knows the whole model, its communication with other agents allows them to take part in minimization of tool to goal error. By combining of reactive and cognitive agents we allow for localized adjustments to environment as well as global goal achievement. Simulation results and conclusions from a planar 4 degrees of freedom model have been presented.**

***Index Terms*—multi agent, redundant manipulator, reactive agents**

I. Introduction

In the main stream of the robotics society, we are taught to view robotic manipulators as singular entity for which is need to design "intelligence" that will process sensory inputs into plans and activate motors to generate motion. In the last 30 years with increasing development of computational units and general increase in computational power, machines needed to compute the behavior and reactions of robots have grown powerful but algorithms are still treating the system as a whole.

Changes in this viewpoint started to come up in [1], where the idea that a control of robotic system can be observed as multi-layer module system that communicates is introduced. This allowed for robots to react locally to obstacles while calculating plans to achieve given goals. Unfortunately this idea was only applied to mobile robots giving rise to automated warehouses and facilities.

Some researches on the other hand have applied distributed system ideas to simplifying inverse kinematic calculations. The problem with inverse kinematics comes with redundancy of robotic system, where inverse of Jacobian matrix between joint states and position and orientation. In [2] an idea was discussed where each joint of robotic manipulator was treated as individual agent/robot that follows simple set of rules. The research showed that it is possible to avoid complex inverse kinematic calculations. This was done by following sequence where last joint turns to goal, calculates position where his base would be to reach the goal and propagates that back to previous joint as its goal. This simple approach successfully avoided the need of complex calculations.

Following from that, [3] built up the system of agents whose actions were no longer needed to be sequential from tool to base. They have allowed for their joint agents to act independently of others, by measuring their own distance to goal. This lead to ability to be fault-tolerant, that is as each regards its own error to goal, if one blocks, others will naturally compensate. This leads to emergent behavior where each joint acts independently of others according to its goals. Advantage that arises is flexibility, but this draws that each agent is capable of same sensory measurements, causing the system to use up more energy. Another disadvantage is consequence of emergence of behavior itself that depends on agents non-flexible behavioral patterns. Thus, a small change on single agent behavior could cause the system not to exhibit same group behavior.

Observing this development, question arises: Can we reduce the number of agents that need sensory input and compensate it with communication network and localized intelligence?

Introducing communication network to set of agents brings in advantages but also design challenges and increase of variables as many books testify ([4], [5]). As it is not the main topic of this paper, we will not explore convergence and network design, but rather just use simple network and models to show how this can be achieved.

II. Models of joints as agents in network

In the following sections different types of agents that communicate in the network will be defined. Their models will be used onwards in research to investigate emerging behaviors for redundant arm manipulators.

A. Model of joint reactive agent

Reactive agents are represented here as pairs of rotational actuator and link that rotates as consequence of actuation. They are designed to be passive receivers in state communication network. As they consist of single rotational joint and link, their forward and inverse kinematics can be easily calculated on board, allowing for fast simple calculations. Calculating the end effector position, marked with orange circle in Fig. 1, with L as link length and q as joint state is as simple as in (1).

$$\begin{bmatrix} x \\ y \end{bmatrix} = L \begin{bmatrix} \cos(q) \\ \sin(q) \end{bmatrix} \tag{1}$$

$$u = K_p e \tag{2}$$

Vuk Vujović is with the School of Electrical Engineering, University of Belgrade, 73 Bulevar kralja Aleksandra, 11020 Belgrade, Serbia (e-mail:vuk.v.vujovic@gmail.com).

Kosta Jovanović is with the School of Electrical Engineering, University of Belgrade, 73 Bulevar kralja Aleksandra, 11020 Belgrade, Serbia (e-mail:kostaj@etf.rs).

Proceedings of 4th International Conference on Electrical, Electronics and Computing Engineering, IcETRAN 2017, Kladovo, Serbia, June 05-08, ISBN 978-86-7466-692-0 pp. ROI1.5.1-4

To control the agent position proportional controller (2) was chosen. Proportional gain K_p is used to map angle error state e to control input u. Achieving smooth transitions would require PID or a more complex controller and can be discussed further.

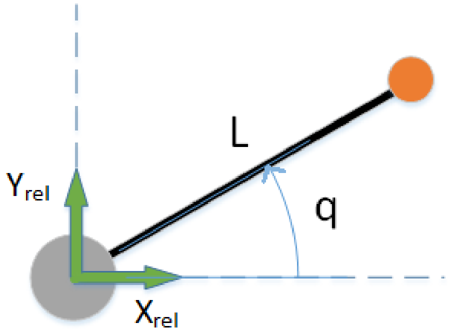

Fig. 1. Parameters and relations in simple reactive agent. Its two main parameters **L** and **q** are shown in local coordinate system of reactive agent.

B. *Model of sensory agent*

Starting from reactive agents, need for agent with sensory input on top arises. This comes in the form of sensory agent, which can measure angle of alignment of its link to goal position, and return angle error as shown Fig. 2. This is heading error, which sensory agent uses to adjust control to keep pointing in the direction of the goal.

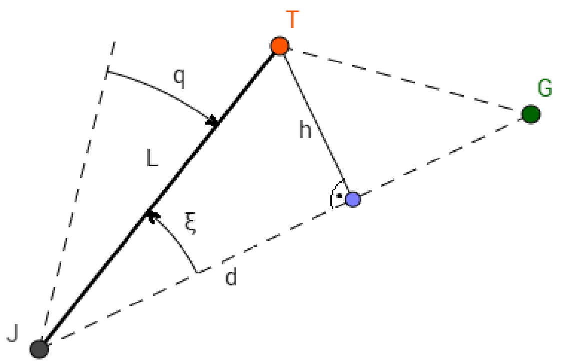

Fig. 2. Representation of heading sensor measurements for sensory agent. Joint **J** with link of length **L** ends with point **T**, when in state **q**. This yields triangle with goal point **G** that is used to measure heading error ξ. Calculations use height **h** of formed triangle.

Based on Fig. 2 we can define equation to calculate heading error as shown in (3).

$$\xi = s(T, \overrightarrow{JG})\,|atan2(h,d)| \tag{3}$$

Where first part of the product is used to determine the sign of the error angle based on side of vector $\overrightarrow{JG}$ where point **T** resides as shown in (4). As expected, this is simple cross product of appropriate vectors.

$$s(T, \overrightarrow{JG}) = \mathrm{sgn}(\overrightarrow{JG} \times \overrightarrow{JT}) \tag{4}$$

This agent will be placed on the top (head) of the chain, and will be referred in use cases as head agent.

C. *Models of base agent*

Given already presented agents in the network, it is still not possible to achieve full position control. Distribution of sensory error through the network will yield directing of manipulator towards the goal, which is as expected consequence of measured error. In this model, instead of introducing another different sensor type, same type of sensor was placed in another type of agent – base agent.

Base agent, apart from being derived from sensory agent is adding a layer of intelligence on top. It is designed to lead the whole manipulator and has twofold goal:

- To bring mechanism in vicinity of goal
- To provide support for head agent in order to achieve tool positioning on top of goal

In order to do this, it switches between two modes of behavior, which represent its intelligence layer and coincide with two goals:

- Setting heading of entire manipulator while blocking communication in network to prevent fine adjustments too early
- Aligning base joint to have goal and tool on same heading and allowing communication in network

III. Model of Whole Redundant Manipulator

Now when we assemble the whole redundant arm manipulator, which is in our case consisting of four agents as in Fig. 3. Second and third agent are reactive agents, connected to network, while **J1** is base agent and **J4** is head agent.

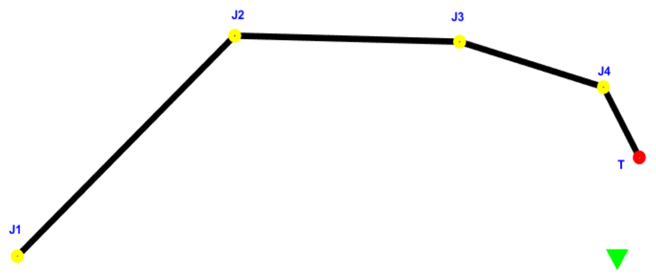

Fig. 3. Representation of whole redundant manipulator assembled from four different agents: **J1** as base agent, **J4** as head agent and **J2** and **J3** as reactive agents. **T** in the image represents gripper or tool as end effector while green triangle is goal position to be achieved.

This model and configuration was used in examples through the paper.

IV. Model of Network of Agents

All of the previously defined models of agents and their connection in kinematic chain to create redundant arm manipulator are connected in communication network to facilitate flow of error and joint states. As we have already defined the behavior of individual agents, Fig 4. shows this specific network where **J1** as base agent influences propagation of information from **Goal** and individual agents onwards.

The information flow from **J4** to **J3** and from **J3** to **J2** in case when it is allowed by **J1** is sign of first derivative of state of previous agent.

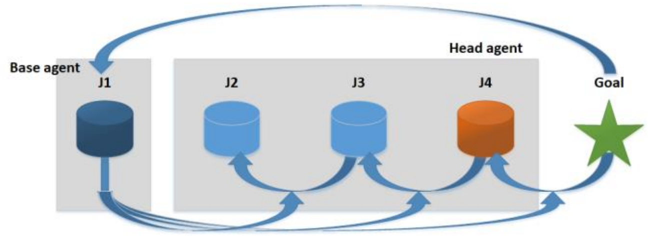

Fig. 4. General representation of information flow in communication network with **J1** base influencing the overall flow.

This allows agents **J2** and **J3** to follow the behavior of the head agent, while still remaining unaware of the entire model of the arm manipulator. This communication step can be represented for agent **k** as in (5).

$$e_k = \mathrm{sgn}\left(\frac{d}{dt}q_{k+1}(t)\right)\Delta \quad (5)$$

Where Δ represents small constant introduced to provoke motion in the agent **k**.

V. Emerging Behavior in Manipulator

Using defined models of agents, redundant manipulator chain and communication network we observed expected behaviors in simulated environment. System behaved as designed, with two main stages of moving to area near goal and fine positioning to goal shown respectively in subsections A and B.

A. Moving to area near the goal

Starting from the configuration shown in Fig. 5 where base agent is connected to the "ground" marked with blue line, and head agent is ending with tool marked with red dot. It can be observed that achieving of the goal position requires adjustments to multiple joints so our algorithm can show full behavior. Also, starting pose is chosen not to be fully extended or collapsed so versatility of the joint agents can be shown.

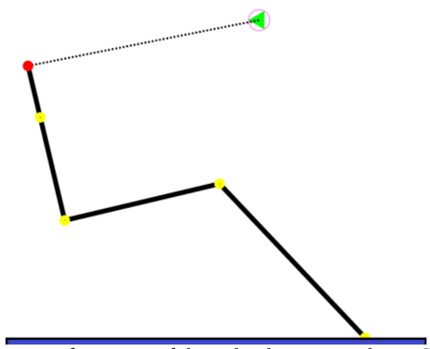

Fig. 5. Starting configuration of the redundant manipulator. Green triangle represents goal position, surrounded with red circle as tolerated area for goal achievement. Dashed line connects tool (end-effector) with goal position to show where does the system strive to go to.

At this stage, base agent blocks communication of other agents and proceeds to fulfill his goal of bringing the chain in the local area of the goal. As mentioned before, this is done by aligning base agent's link with line-of-sight with goal.

Once this goal is achieved with certain numeric tolerance, configuration shown in Fig. 6 is result. We can see that no other agent has actuated its joint, and from this point onward communication in network is active allowing head agent to fine position tool at the goal.

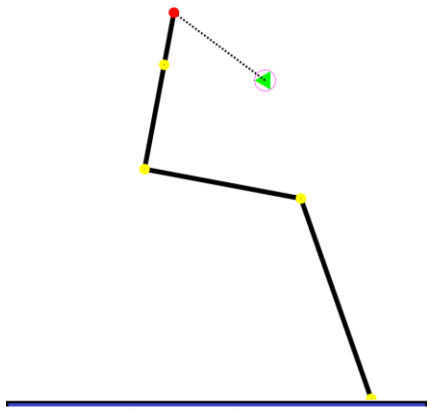

Fig. 6. Local goal area configuration of the redundant manipulator. Tool is now in better position to approach the goal.

B. Moving to area near the goal

After base agent's first goal is finished, communication is active and head agent rotates towards the goal. This information is propagated through the network back to all reactive agents. They start bending to allow for smoother curvature of the whole structure.

In order to ensure achieving of the goal position, base agent adjusts its state to have goal and tool in line-of-sight. This, due to absence of distance error enforces reaching of the goal state. Final state with tool at goal position is shown in Fig. 7.

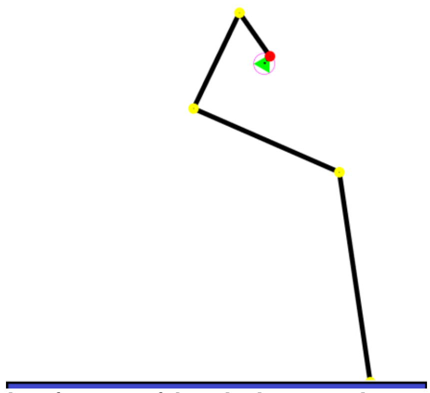

Fig. 7. Final configuration of the redundant manipulator upon achieving the goal position. Due to simulation limitations, goal position is accepted with tolerance radius shown with red circular line.

VI. Discussion and Drawbacks

This approach shows alongside [2] and [3] that there is no need for full knowledge of the model in order to perform kinematic control of redundant chains, but carries drawbacks as well.

While [2] uses sequential algorithm heading from tool to base, this approach is operating in distributed manner, allowing for more flexible solutions, and reduces execution dependency in control system. On the other hand, approach shown here has sequential parts in form of two goals of the base agent, preventing to declare it as fully distributed algorithm.

Idea shown in [3] gives great example of absolutely distributed method, albeit at the price of having each agent equipped with sensor. This can prove to be costly thing if methods are to be applied to chains with many degrees of redundancy. To counter that, our approach uses sensors in only two agents – base and head, yielding better energy and cost efficiency for applications.

There are of course, drawbacks and pitfalls to presented method as well. One of them is the sensor measurement in base agent that can switch angle measurement depending on the task. In practical applications, this would mean a depth camera with image processing module or similar sensor.

VII. Conclusion and outlook

This paper shows concept of a new partially distributed method of redundant arm manipulator control. It extends on existing research by reducing number of sensory inputs via replacement with communication network among individual agents. In the future, different more complex and versatile networks could be explored where simple rules could yield complex emerging behaviors. Apart from that, there is space for establishing mathematical proofs and foundations for these algorithms.

As an application of this research area we see

Acknowledgment

This work was partly funded by the Ministry of Education, Science and Technological Development, Republic of Serbia, under contracts TR-35003.

References

[1] R. Brooks, "A Robust Layered Control System For a Mobile Robot", MIT, Cambridge, USA, 1985

[2] D. Duhaut, "Using multi agent approach to solve inverse kinematic problem," Proceedings of IROS, Yokohama, Japan, pp. 2002-2007, July 1993

[3] S. Delarue, P. Hoppenot, E. Colle, "A multi agent controller for a mobile arm manipulation," ICINCO, Angres, France, pp. 292-299, May 2007

[4] M. Mesbahi, M. Egerstedt, *Graph theoretic methods in multiagent networks*, 1st ed. Princeton, United States: PUP, 2010

[5] M. Zhu, S. Martinez, *Distributed Optimization-Based Control of Multi-Agent Networks in Complex Environments*, 1st ed. Springer, United States: Springer, 2015