Proc. 3rd International Symposium on Methods and Models in Automation and Robotics MMAR'96, 10-13 September 1996, Międzyzdroje, Poland, Vol.3, pp.893–898

REACTIVE ROBOT CONTROL APPLIED TO ACQUIRING MOVING OBJECTS[‡]

CEZARY ZIELIŃSKI*

*Warsaw University of Technology, Institute of Control and Computation Engineering, ul. Nowowiejska 15/19, 00-665 Warsaw, POLAND, C.Zielinski@ia.pw.edu.pl

Abstract. The paper presents a formalized approach to reactive robot control. Reactive robot control consists in actively using sensor readings to modify robot goal pursuing actions [1, 2, 3, 4]. This control method is ilustrated by an example of a robot acquiring moving objects from a conveyor.

Key Words. Reactive control; Robot programming; Robot control

1 INTRODUCTION

Utilization of information obtained from external sensors of a robot to control its motion has been the subject of research for several years now [5]. The ongoing research concentrates on: development of new types of robot sensors (e.g., [6]) and incorporation of many types of external sensors into a robotic system (e.g., [7]). The problems of data aggregation (fusion) are considered too (e.g., [8, 9]). Robot control systems relying on sensor data take different approaches to the problems of: integration of multiple sensors into a robotic system (e.g., [10, 11, 12, 13]), formal task description for robots equipped with sensors (e.g., [14, 15]), sensor data aggregation and interpretation (e.g., [8, 9]). A comprehensive discussion of some of the above topics can be found in [9].

Artificial intelligence approach to robot control strongly relies on world models to execute a task. Sensors, in this case, are mainly used to update the world model, which in turn is used in the generation of the plan of actions. On the other hand, behavioural control concept does not need a world model to execute a task [16, 17, 18]. In this case the controller is built of several finite state automatons functioning in parallel, each achieving a single objective by a certain behaviour. The controller is constructed incrementally by adding ever more complex layers of behaviours on top of the more elementary ones. Upper layers examine data from lower levels and can suppress or inhibit their behaviours.

In this paper the goal that is to be achieved and a single layer of actions (that can also be called behaviours or reactions) are distinguished. These actions are triggered when sensors detect appropriate conditions. Unlike the pure behavioural approach, where the partitioning of the system is intuitive, a formal path was followed.

1.1 Theoretical discussion

A robotic system is decomposed into three subsystems: *effectors* (manipulator arm or arms, tool and the cooperating devices), *receptors* (*real sensors*), and the *control subsystem* (e.g., memory). The state $s \in S$ of such a system can be denoted in the following way:

$$s = < e, r, c >, \qquad s \in S, \quad e \in E, \quad r \in R, \quad c \in C, \tag{1}$$

where:
e is the state of the effectors,
E is the effector state space,
r is the state of the real sensors,
R is the real sensor reading space,
c is the control subsystem state,

[‡] The project has been supported by the statutory funds of Warsaw University of Technology.

C is the control subsystem state space.

The raw data obtained from real (hardware) sensors usually cannot be utilized directly to control the system. It has to be transformed into a useful form. This transformation is called *data aggregation.* As a result of this a *virtual sensor reading* v is obtained.

$$v = f(r), \quad v \in V \tag{2}$$

Vector function f is called an *aggregating function.* V is the virtual sensor reading space. In more complex cases the aggregating function (2) can take the form $v = f(r, c, e)$, e.g. when the real sensors are mounted on a moving end-effector and when the final virtual sensor reading depends on the history of previous real sensor readings.

First, data obtained from real sensors is aggregated into a virtual sensor reading. Next the virtual sensor reading space is partitioned into disjoint subspaces V_j, $j = 0, \ldots, j_R$, where $j_R + 1$ is the number of these subspaces:

$$\begin{aligned} V &= V_0 \cup \bigcup_{j=1}^{j_R} V_j, \quad \text{and} \quad \forall_{j \neq q} V_j \bigcap V_q = \emptyset, \\ & \quad q = 0, \ldots, j_R, \end{aligned} \tag{3}$$

With each of the subspaces V_j a reaction template B_j, $j = 0, \ldots, j_R$, is associated, where $j_R + 1$ is the number of subspaces and so also the number of reaction templates. V_0 is the neutral virtual sensor reading subspace and B_0 is the main goal pursuing reaction associated with it. A reaction instance generated by B_0 is executed whenever no other reaction is realised and the virtual sensor readings remain within the neutral subspace V_0. If during the realisation of the goal, the sensor readings "enter" a subspace $V_j \subset V$ associated with a reaction template B_j, then the realisation of the global goal or any other reaction is interrupted and a reaction instance b_j of B_j is triggered. In other words, virtual sensor readings $v_j \in V_j$ trigger the reaction $b_j \in B_j, j = 1, \ldots, j_R$. Reaction instance b_j is executed as a sequence of steps.

$$b_j = (e_j^0, c_j^0) \ (e_j^1, c_j^1) \ \ldots \ (e_j^T, c_j^T), \ \ b_j \in B_j, \tag{4}$$

where (e_j^0, c_j^0) is the state of the system in which reaction instance b_j is initiated (this is also the state in which the preceding reaction has been interrupted); (e_j^T, c_j^T) is the terminal state of the system after the reaction b_j ends its execution.

2 EXAMPLE: ACQUIRING MOVING OBJECTS

One of the tasks chosen for the presentation of reactive control concept consisted in impactless grasping of objects moving on a conveyor. It was assumed that the objects are cylinders standing vertically with their circular bases parallel to the plane of the conveyor. The localisation of a cylinder and the correction of the robot gripper position were based on the information acquired by infra-red (IR) barriers and proximity sensors. It was assumed that the conveyor moves with approximately constant speed w. This velocity is known, because the control system itself commands it. The approximate grasping position was anticipated from the following relationships (Fig.1):

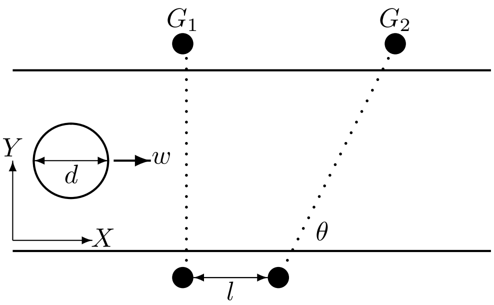

Fig. 1 The method of measuring the position of an object moving on a conveyor

$$\begin{cases} x = x_{G_1} + w(t - t_{p_1}) - \frac{d}{2} \\ y = [w(t_{p_2} - t_{p_1}) - l] \tan\theta \\ z = h_g \end{cases} \tag{5}$$

where: x, y, z are the coordinates of the current position of the cylinder grasping location in relation to the local coordinate frame affixed to the conveyor (z does not change as the height of the cylinder grasping location h_g above the conveyor is constant), θ is the angle between the second IR-barrier and the conveyor motion direction, l is the shorter distance between the IR-barriers measured along the conveyor edge, t is the current time, t_{p_1} and t_{p_2} are the instants at which the leading edge of a cylinder is detected by the two consecutive IR-barriers and x_{G_1} is the distance of the first IR-barrier (G_1) from the local coordinate frame measured along its X axis.

The signals obtained from the two IR-barriers are presented in Fig.2. These signals and the knowledge of conveyor velocity suffices to determine the cylinder diameter $d = w(t_{k_1} - t_{p_1})$, and thus the necessary distance between the gripper jaws $d + \varepsilon$, where ε is a small extra distance enabling collision free embrace. The knowledge of $x(t)$, y, z and d is sufficient to calculate a convenient instant and location for grasping the object. It is assumed that the relationship between the robot reference coordinate frame and the local coordinate frame affixed to the conveyor is known, and so, for the purpose of this example, all coordinates can be expresses in the local frame (Fig.1).

Due to disturbances, inaccuracies of measurements obtained from the IR-barriers and lack of

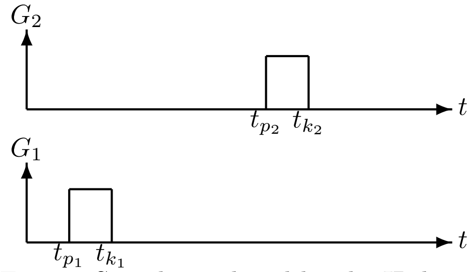

Fig. 2 Signals produced by the IR-barriers G_1 and G_2 when an object moving on the conveyor encounters them

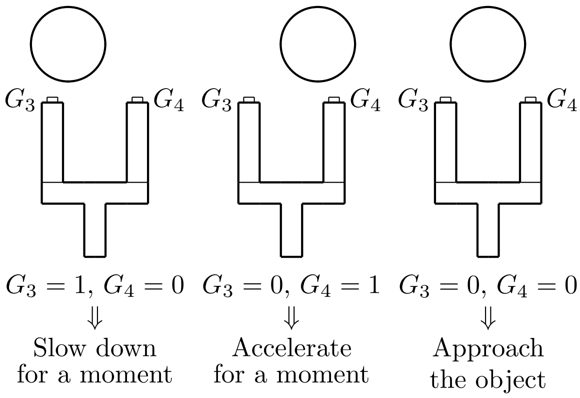

Fig. 3 Correction of the gripper motion

precise calibration of the system (e.g. the transformation between the robot and local coordinate frames is known approximately) a final adjustment of gripper motion is necessary prior to the grasping phase. This is accomplished by reactive control. For that purpose the gripper fingers are equipped with two IR proximity sensors G_3 and G_4, which detect the presence of obstacles in the approach path to the cylinder grasping location, i.e. the object misplaced in relation to the anticipated position. The concept of the correction is presented in Fig.3. The two real sensors G_3 and G_4 can be treated as a single virtual sensor v supplying a two bit reading from the set: $\{00, 01, 10, 11\}$. This is a very simple form of data aggregation: $v = f(G_3, G_4)$. The virtual reading space is also trivial – it has four elements: $\{00, 01, 10, 11\}$ and so it was partitioned into four one-element subspaces.

With each of the four subspaces an adequate robot reaction is associated.

$$\begin{cases} V_{00} : & v = 00 \Rightarrow B_j, & \text{– continue} \\ & j = 00, 01, 10, 11 & \\ V_{01} : & v = 01 \Rightarrow B_{01} & \text{– accelerate} \\ V_{10} : & v = 10 \Rightarrow B_{10} & \text{– slow down} \\ V_{11} : & v = 11 \Rightarrow B_{11} & \text{– error} \end{cases} \tag{6}$$

When the virtual sensor reading belongs to the neutral subspace V_{00} the currently executed reaction is continued until its termination and then the main reaction B_{00} (i.e. the task goal pursuing reaction) is invoked. Whenever the virtual sensor reading enters one of the subspaces: V_{01}, V_{10}, V_{11} the currently executed reaction is aborted and the reaction: B_{01}, B_{10} or B_{11} respectively is invoked. The state of the effectors e has three components: state of the conveyor e_C, state of the robot e_R and state of the gripper e_G.

The definitions of reactions: B_{00}, B_{01}, B_{10}, B_{11} are formulated in terms of sequences of states of effectors and control subsystem. Virtual sensor readings are treated as an input to the so reduced system. The definition of the main reaction B_{00} is:

$$b_{00} \in B_{00}, \quad b_{00} = (e^0_{00}, c^0_{00}), (e^1_{00}, c^1_{00}) \ldots (e^{i_T}_{00}, c^{i_T}_{00})$$

$$\begin{cases} c^0_{00} & : \; c^0 \\ e^0_{00} & : \; e^0_C,\ e^0_R,\ e^0_G \\ c^i_{00} & : \; \Delta e_C = w\Delta t, \\ & \quad\ \Delta e_{Rx} = w\Delta t, \\ & \quad\ \Delta e_{Ry} = \frac{d}{i_T - 2} \\ e^i_{00} & : \; e^i_C = e^{i-1}_C + \Delta e_C, \\ & \quad\ e^i_{Rx} = e^{i-1}_{Rx} + \Delta e_{Rx}, \\ & \quad\ e^i_{Ry} = e^{i-1}_{Ry} + \Delta e_{Ry} \\ c^{i_T-1}_{00} & : \; \Delta e_C = w\Delta t, \Delta e_{Rx} = w\Delta t \\ e^{i_T-1}_{00} & : \; e^{i_T-1}_C = e^{i_T-2}_C + \Delta e_C, \\ & \quad\ e^{i_T-1}_{Rx} = e^{i_T-2}_{Rx} + \Delta e_{Rx}, \\ & \quad\ e^{i_T-1}_G = d \\ c^{i_T}_{00} & : \; \Delta z = h_l \\ e^{i_T}_{00} & : \; e^{i_T}_{Rz} = e^{i_T-1}_{Rz} + \Delta z \end{cases} \tag{7}$$

where: $i = 1, \ldots, i_T - 2$ – the main reaction discrete time instants with a period (step) Δt, c^0 – control subsystem state, e^0_C – conveyor state, e^0_R – robot state, and e^0_G – gripper state, all at an instant when the main reaction begins. The initial state of the execution of the main reaction has to fulfil the following conditions:

$$\begin{aligned} e^0_{Rx} &= x_{G_1} + w(t - t_{p_1}) - \tfrac{d}{2}, \\ &\quad \text{where } t \text{ is the current time } t > t_{k_2}, \\ e^0_{Ry} &= [w(t_{p_2} - t_{p_1}) - l] \tan\theta - \tfrac{d}{2}, \\ e^0_{Rz} &= h_g, \\ \dot{e}^0_{Rx} &= w, \\ \dot{e}^0_{Ry} &= 0, \\ \dot{e}^0_{Rz} &= 0, \\ \dot{e}^0_C &= w, \\ e^0_G &= d + \varepsilon. \end{aligned}$$

In a similar fashion the constant gripper orientation, included in e_R, can be specified. In the definition of the consecutive steps the constant components have been omitted. Whenever the virtual sensor readings leave the neutral subspace the main reaction is interrupted and a corrective reaction is invoked. When the corrective reaction terminates and the virtual sensor reading is within the neutral subspace the main reaction continues its execution. If the corrective reaction does not drive the virtual sensor reading into the neutral subspace other reactions are invoked until the reading is forced into the neutral subspace.

The goal of the main reaction is to drive the robot gripper in the X direction at the same speed as the velocity of the cylinder moving on the conveyor and at the same time approach it in the Y direction, keeping the Z coordinate and the gripper orientation constant. In its last two steps the gripper is closed and the cylinder lifted above the conveyor surface. To simplify the presentation the system halts at that instant.

Unlike the main reaction, the corrective reactions always commence from their initial step 0 and either terminate executing all of their prescribed steps or are aborted because of the virtual sensor reading straying into other reaction's virtual sensor subspace. Because B_{01}, B_{10} and B_{11} are single step reactions they always terminate without interruption.

$$b_{01} \in B_{01}, \quad b_{01} = (e^0_{01}, c^0_{01}) \ (e^1_{01}, c^1_{01})$$

$$\begin{cases} c^0_{01} & : \ c^0 \\ e^0_{01} & : \ e^0_C, \ e^0_R, \ e^0_G \\ c^1_{01} & : \ \Delta e_C = w\Delta t, \ \Delta e_{Rx} = \delta w \Delta t \\ e^1_{01} & : \ e^1_C = e^0_C + \Delta e_C, \\ & \quad e^1_{Rx} = e^0_{Rx} + \Delta e_{Rx} \end{cases} \tag{8}$$

The goal of this reaction is to accelerate during one step, hence $\delta > 1$. The definition of reaction $b_{10} \in B_{10}$ is similar to (8), but $\delta < 1$ to decelerate the motion of the gripper. The value of δ depends on ε.

In the case when the detected object has a too large a diameter, the conveyor is stopped, the gripper raised and the system halts.

$$b_{11} \in B_{11}, \quad b_{11} = (e^0_{11}, c^0_{11}) \ (e^1_{11}, c^1_{11})$$

$$\begin{cases} c^0_{11} & : \ c^0 \\ e^0_{11} & : \ e^0_C, \ e^0_R, \ e^0_G \\ c^1_{11} & : \ \Delta e_{Rz} = h_l \\ e^1_{11} & : \ e^1_C = e^0_C, \ e^1_{Rz} = e^0_{Rz} + \Delta e_{Rz} \end{cases} \tag{9}$$

Obviously other action could be taken, e.g. the distance between the gripper jaws could be increased.

The fundamental advantage of the above formal specification is the easiness of transformation of this specification into a `C` language program controlling the acquisition of moving objects from the conveyor by the robot. Here `pseudo-C` is used, to make the transformation from the specification to the resulting code evident. `Pseudo-C`, because the identifier names do not fulfil the `C` syntax rules (the names of variables and constants have been retained as in the reaction definitions).

Specification (6) and reaction (7), is transformed into a `switch` statement with the condition depending on the virtual sensor reading.

```
.   .   .   .   .
// The current state of the robot is:
// e_R[x] = x_{G_1} + w * (t - t_{p_1}) - d/2
// e_R[y] = (w * (t_{p_2} - t_{p_1}) - l) * tan(θ)
// e_R[z] = h_g
int i = 0;
for (;;) {
   read_virtual_sensor(v);
   switch (v) {
     case 00:
       Δe_C = w * δt;
       Δe_R[x] = w * δt;
       Δe_R[y] = d/(i_T - 2);
       move_robot_and_conveyor(Δe_R, Δe_C);
       break;
     case 01:  B_01(w, Δt); continue;
     case 10:  B_10(w, Δt); continue;
     case 11:  B_11(w, Δt); exit;
   }; // end:  switch (v)
   if (i < i_T - 1)
     i++;
   else
     break;
}; // end:  for (;;)
e_G = d;
close_gripper (e_G);
Δe_R[z] = h_l;
move_robot_and_conveyor(Δe_R, 0);
halt;
.   .   .   .   .
```

The specification of reactions B_{01} and B_{10} (8) and B_{11} (9) is transformed into procedures executing the described steps. The code executing reaction B_{01} (8) is the following. The code executing the other two reactions is similar.

```
void B_01 (double w, double Δt) {
   double Δe_C;
   double Δe_R[6] = {0,0,0,0,0,0};
   const δ = 1.1;
   Δe_C = w * Δt;
   Δe_R[x] = δ * w * Δt;
   move_robot_and_conveyor (Δe_R, Δe_C);
}; // end:  B_01()
```

The above transformations are described in [2, 3, 4] and the implementation details of this example in [19].

3 CONCLUSIONS

The main advantage of reactive robot control is its formal specification which greatly simplifies the task of coding the program and the fact that it can be adapted to sensors of any complexity and quantity by using the concept of virtual sensors. It is very well suited to discrete event tasks, but its usefulness in continuous control problems depends on the sampling time of the servo-controllers.

As the sensor reading space, robot reactions and the global goal can be described formally, a formal specification of the robotic controller realising a given task is obtained. This specification is later used as the basis for coding the software of a controller tailored to the needs of the execution of the specific task. The program executing the task is implemented on a Multi-Robot Research-Oriented Controller – MRROC [3, 20, 21]. Once the reactions have been specified formally the implementation is straight forward and to a large extent could be done automatically, i.e. the control program could be generated by a pre-processor or a compiler.

The other advantage of reactive control concept is incremental controller design method. The virtual sensor reading space can be initially divided into few coarse subspaces and hence only few reactions have to be coded. If the so obtained controller does not behave properly, some of the subspaces can be further divided and new reactions can be added. The structure of the controller does not change only new reaction procedures are supplied and new `case` statements are added to the `switch` instruction. In this way the functioning of the controller can be fine-tuned experimentally.

REFERENCES

[1] C. Zieliński, T. Zielińska: *Sensor-Based Reactive Robot Control*, Proc. 10th CISM--IFToMM Symposium on Theory and Practice of Robots and Manipulators, Ro.Man.Sy'94, 12-15 September 1994, Gdańsk, Poland, Ed. A. Morecki and G. Bianchi and K. Jaworek, Springer-Verlag, 1995.

[2] C. Zieliński: *Reaction Based Robot Control*, Int. J. of Mechatronics, Vol. 4, No. 8, 1994, pp. 843-860.

[3] C. Zieliński: *Robot Programming Methods*, Publishing House of Warsaw University of Technology, 1995.

[4] C. Zieliński: *Sensorimotor robot control*, Proc. 7th IFAC/IFORS/IMACS Symposium on Large Scale Systems: Theory and Applications, 10–13 July 1995, London, United Kingdom.

[5] J. Pinkava: *Towards a Theory of Sensory Robotics*, Robotica, Vol. 8, 1989, pp. 245-256.

[6] S.R. Ruocco: *Robot Sensors and Transducers*, University Press, 1987.

[7] E. Grant: *Uncertainty in Robot Sensing*, Ed. C.S.G. Lee, Springer-Verlag, NATO ASI Series F, Vol. 66, *Sensor-Based Robots: Algorithms and Architectures*, 1991, pp. 41-57.

[8] R.C. Luo, M-H. Lin, R.S. Scherp: *Dynamic Multi-Sensor Data Fusion System for Intelligent Robots*, IEEE J. of Robotics and Automation, Vol. 4, No. 5, August, 1988, pp. 386-396.

[9] : R.C. Luo, M.G. Kay: *Multisensor Integration and Fusion in Intelligent Systems*, IEEE Trans. on Systems, Man, and Cybernetics, Vol. 19, No. 5, September/October, 1989, pp. 901-931.

[10] Y.F. Zheng: *Integration of Multiple Sensors into a Robotic system and its Performance Evaluation*, IEEE Trans. on Robotics and Automation, Vol. 5, No. 5, October, 1989, pp. 658-669.

[11] U. Rembold, P. Levi: *An Integrated Sensor System for Robots*, Ed. C.S.G. Lee, Springer-Verlag, NATO ASI Series F, Vol. 66, *Sensor-Based Robots: Algorithms and Architectures*, 1991, pp. 3-24.

[12] Y. Wang, S. Butner: *RIPS: A Platform for Experimental Real-Time Sensory-Based Robot Control*, IEEE Trans. on Systems, Man, and Cybernetics, Vol. 19, No. 4, July/August, 1989, pp. 853-860.

[13] C. Zieliński: *Controller Structure for Robots with Sensors*, Int. J. of Mechatronics, Vol. 3, No. 5, 1993, pp. 671-686.

[14] D.M. Lyons, M.A. Arbib: *A Formal Model of Computation for Sensory-Based Robotics*, IEEE Trans. on Robotics and Automation, Vol. 5, No. 3, June, 1989, pp. 280-293.

[15] D.M. Lyons, M.A. Arbib: *A Task-Level Model of Computation for Sensory-Based Distributed Control of Complex Robot Systems*, Poceedings of the Symposium on Robot Control, Barcelona, Spain, November, 1985.

[16] R.A. Brooks: *Intelligence Without Representation*, Artificial Intelligence, Vol. 47, 1991, pp. 139-159.

[17] R.A. Brooks: *A Robust Layered Control System For a Mobile Robot*, IEEE J. of Robotics and Automation, Vol. 2, No. 1, March, 1986, pp. 14-23.

[18] J.H. Connel: *A Behavior-Based Arm Controller*, IEEE Trans. on Robotics and Automation, Vol. 5, No. 6, December, 1989, pp. 784-791.

[19] Leśkiewicz P.: *Utilisation of proximity sensors in robot control*, M.Sc. Thesis supervised by C. Zieliński, Warsaw University of Technology, Institute of Control and Computation Engineering (in Polish), 1995.

[20] Szynkiewicz W., Zieliński C., Kierzenkowski K., Zielińska T., Grodecki A.: *Multi-Robot Research-Oriented Controller MRROC*, PATIA Memo, Warsaw University of Technology, (in Polish), May, 1995.

[21] C. Zieliński: *Control of a Multi-Robot System*, Proc. 2nd International Symposium on Methods and Models in Automation and Robotics MMAR'95, 30th August – 2nd September 1995, Międzyzdroje, Poland.