Cloud-Based Digital Twin for Robot Integration in Intelligent Manufacturing Systems

Florin Anton, Theodor Borangiu(✉), Silviu Răileanu, and Silvia Anton

Department of Automation and Applied Informatics, University Politehnica of Bucharest, Bucharest, Romania
{florin.anton, theodor.borangiu, silviu.raileanu, silvia.anton}@cimr.pub.ro

Abstract. The paper describes the architecture design and implementing solution for the digital twins of industrial robot, aggregated and embedded in the global health monitoring, maintenance and control system of manufacturing resources. Manufacturing scheduling and control system. The main functionalities of the digital twin are: monitoring the current status and quality of services performed by robots working in the shop floor, early detecting anomalies and unexpected events to prevent robot breakdowns and production stops, and forecasting robot performances and energy consumption. Machine learning techniques are applied in the cloud layer of the virtual twin for predictive, customized maintenance and optimized robot allocation in production tasks. The paper introduces a framework integrating the virtual robot twins in an ARTI-type control architecture, proposes a solution to implement the twin on a distributed cloud platform and exemplifies the concepts in a shop floor case study with SCARA assembly robots.

Keywords: Digital twin · ARTI · Industrial robot · Anomaly detection · Predictive maintenance · Edge computing · Cloud computing

1 Introduction

The term 'Industry of the Future' (IoF) indicates the new industrial revolution initiated by a new generation of manufacturing systems designed to be adaptive, fully connected, highly efficient (optimized) and robust (fault tolerant, highly available). The global IoF model describes a new stage of manufacturing fully automatized and robotized, using ever more advanced information, control and communication technologies (IC^{2}T). IoF works with 'intelligent factories', characterized by digitalization and interconnection of distributed manufacturing entities in a 'system of systems' approach: i) new types of production resources (e.g., intelligent robots) highly interconnected and self-organizing in the shop floor, products deciding upon their own work route and resources; ii) smart decisions taken from real-time production data collected from resources and products and processed with machine learning techniques [1].

S. Zeghloul et al. (Eds.): RAAD 2020, MMS 84, pp. 565–573, 2020.
https://doi.org/10.1007/978-3-030-48989-2_60

The digital transformation of manufacturing through resource, product and order virtualization relies on techniques allowing for intelligent decision-making complying with the principles of flexibility and reality-awareness. These issues are addressed in the literature through three domains: big data analytics, machine learning and digital twins [2].

The holistic view of the real capabilities, status and features of an entity (e.g., robot, process, product) including its digital representation, execution context, history of behaviour, time evolution and status can be encapsulated in a **digital twin** (DT) defined as an extended virtual model of the robot, process, product, that is persistent even if its physical counterpart is not always on line/connected; this extended digital model can be shared in a cloud database with other shop floor entities [3]. Robot virtualization relates to the capacity of creating robot services in the cloud – a part of the Cloud Manufacturing (CMfg) technology [4]. Manufacturing control frameworks like Cyber-Physical Production Systems (CPPS) that use machine learning (ML) in CMfg models.

The idea of DT is currently being introduced in manufacturing control architectures to virtualize three abstract entity classes: resources (robots, machines), products (client preferences, recipes,) and orders (operation sequences, precedencies). It becomes a key element between asset role and decision making and is able to monitor, forecast and influence the behaviour of the physical twin. Two main classes of DT are of interest in robotized manufacturing [6]:

- Digital twin of resources and products: a) monitoring the health, quality of services (QoS) performed by resources, detecting anomalies in their behaviour and launching predictive maintenance programs; b) performing the traceability of products.
- Digital Twin of organizations: mirroring and forecasting the control system's actions and performances for batch optimization, robustness and process resilience.

The paper introduces a framework of a reality-aware digital twin for industrial robots integrated in intelligent manufacturing systems that use cloud services performed on two layers: i) a layer distributed among fog computing elements, and ii) a centralized cloud IaaS (Infrastructure as a Service). This framework enables robot virtualization, health monitoring and anomaly detection, and the coupling between behavioural robot models and multi-physical processes for real-time predictive robot maintenance and optimal allocation in production tasks based on energy consumption and QoS forecasts.

The rest of the paper is organised as follows: Sect. 2 defines the multi-layered DT architecture for an industrial robot integrated in manufacturing, and the embedding principles. Section 3 describes the edge- and fog computing layers that implement the distributed DTs in multi-robot manufacturing systems. Section 4 presents the Cloud IaaS for the high-level DT layers embedded in predictive individual robot maintenance and collective optimized robot allocation tasks. Experimental results are presented and a set of conclusions are formulated.

2 Digital Twin Architecture of Robots in Manufacturing

The DTs of industrial robots, components of manufacturing systems, are embedded in the global resource management and batch scheduling programs with the following objectives: safe and efficient utilization of robotic resources and optimization of batch production (energy consumption, make-span, balanced resource usage, etc.).

Figure 1 shows the 4-layer DT architecture of the Industrial Robot – component of an intelligent manufacturing system. The lower level Digital Twin layers (DT I and DT II) define the composition of the robot components and processes served by the robot, while the higher level ones (DT III and DT IV) implement methods for predictive robot maintenance and allocation in manufacturing tasks being executed through high performance computing in the cloud:

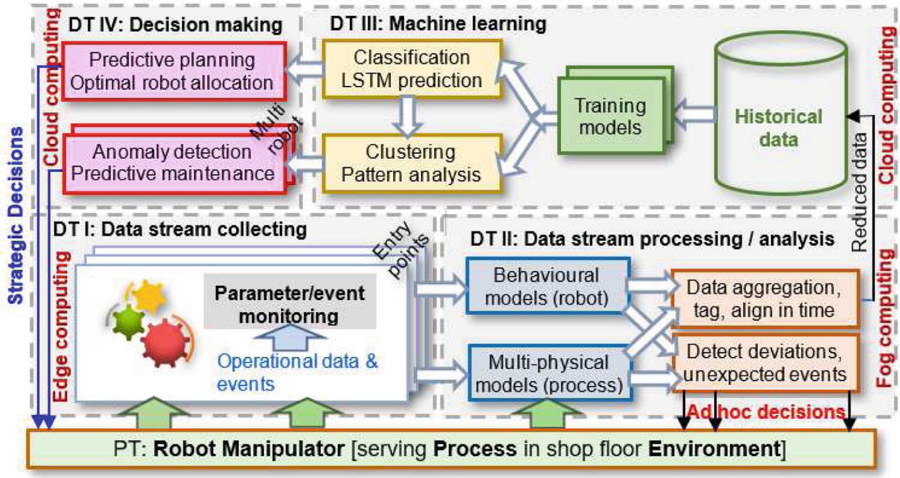

Fig. 1. 4-layer aggregate digital twin embedded for predictive anomaly detection, maintenance and allocation of industrial robots in manufacturing operations

DT I - Data Stream Collecting. This DT element is connected to the physical twin (the components of: the robot manipulator - arm/gripper/tools; robot controller - electronic boards/backplane; machine vision - video cameras/lighting system; external sensors - vibrations/effort/temperature), to the process it serves and to the environment in which it operates, and gathers data from a multitude of shop floor entry points.

DT II - Data Stream Processing and Analysis. The functions of this DT element are: sorting and aligning the data streams generated by DT I in normalized time intervals, and aggregating them using covariance techniques. There are four functional modules composing this second DT element:

- *Repository of behavioural robot models*: it contains models of the robot operating modes that must be defined and called in order to infer correctly the various possible behaviours of the robot system (e.g., full automated/semi-automated mode, dry run etc.). In addition, some behaviour parameters must be defined off-line and eventually re-evaluated at run time (e.g., parameters of the robot's TOOL transformation).

- *Repository of multi-physical process models*: it contains models of the physical processes upon which the robot acts; the parameters of these models depend strongly on the robot operating parameters.
- *Time alignment and data aggregation*: the main function consists in joining raw data streams tagged according to the two above models in application-specific streams. Distributive join/merge big data operations are executed in parallel on multiple worker nodes in map-reduce clusters. Once the information needed for an application is assembled in a 'joined' stream, the next operation is a 'reduce' type one. Multiple aggregations are possible using different keys for map-reduce, i.e. tagging messages by their shop floor location or energy consumption area or heat distribution.
- *Detection of deviations and unexpected events*: this functional component of the DT generates immediate decisions of resource blocking in case unexpected events or significant deviations of the sensed robot parameters from the limit values specified in the behavioural models or from the controller's planned commands are detected.

DT III - Machine Learning. This high-level DT is used to extract insights in real-time from data streams aggregated at the robot workplace (resource, environment, process). For multi-robot shop floors, the computation is parallelized for all robot channels. Three machine learning (ML) techniques are used [5]: *prediction* with deep learning neural network to learn patterns and variations in measurements (e.g., consumed energy); the neural network is then used to make forecasts for these measurements; *classification* - the DT tries to find a class for every feature vector received; *clustering* - the DT tries to find similarities in non-labelled, multidimensional data and tag each feature vector accordingly. The usage of ML needs historical data storage to train the ML models.

DT IV- Intelligent Decision Making. The decision support available on this DT layer allows embedding the robot's networked digital twin in two scenarios: i) weighting the allocation of resources to operations on products in multi-robot manufacturing systems, based on forecasted values of QoS, performances and/or power consumption, for global optimization; ii) anomaly detection and customization of robot maintenance programs: the classification is done at multiple levels; one classification is the robot health using multiple combined KPIs in the map-reduce phase.

3 Collecting and Processing Robot Data at Shop Floor Edge

In order to connect robot sensors, actuators or embedded control devices in an Industrial IoT (IIoT) framework, a dedicated type of hardware was developed – the *IoT gateway* [7]. This hardware connects devices to a network which is in most of the cases directly connected to Internet. Additionally, IoT gateways pre-process and filter the data from shop floor devices locally, at the edge of the CMfg, reducing thus the amount of data to be sent in the Internet with big impact on response time and network transmission costs.

Apart the use of IoT gateways as concentrators of legacy devices and the access to local networks, their processing power can be used to treat more complex information originating from the robot workstation, locally and faster without overloading the cloud. Thus, decision algorithms based on high performance mathematical computation which are not available in the robot operating system can be executed on the IoT gateway and the result returned to the robot controller or to the cloud. These types of algorithms are usually associated with vision systems and with end-effector tools the control part of which is not embedded in the robot controller (in most cases these are dedicated/one of a kind systems built 'in house' by the application integrator).

An example of such robot gripper device is a magnetic tool with collision detection capability. In order to analyse in real-time the data streams generated by robot sensors/external devices supporting multiple communication protocols for data transfer, the IoT gateway was extended in the present research to the aggregation node concept [8]. An aggregation node consists of a group of entities connected to an industrial processing unit – the node's kernel (PC workstation, Single Board Computer - SBC, Next Unit of Computing - NUC) that is flexible enough to run big data software. A NUC is a small form factor computer kit manufactured by Intel [intel.com] which offers the same processing power of desktop PCs. The NUC uses high-end micro-processors and is usually integrated in robot-vision applications since it can be mounted near the image acquisition devices, minimizing thus the processing times due to the direct, high-speed connection. Having standard USB ports, it can be extended with modules for industrial communication and digital/analogue I/O channels.

The following data stream processing of robot parameters are considered for the aggregation nodes on level II of the digital twin:

- Motor temperatures, which joined with the energy consumption data are used for fault detection and preventive maintenance, Fig. 2, A.
- Energy-related parameters for the entire robot manipulator and its n joint motors (voltage, current, phase), Fig. 2, A. This information is collected continuously and per-operation and used in resource scheduling, parameter adjustment (maximum speed and acceleration) and fault detection [9].
- Joint values in order to compute motor utilisation and associate the values with cruise speed, acceleration and payload, Fig. 2, D.
- Additional communication modules for bidirectional interaction between the aggregation node processor and the robot controller in order to map gripper/tool information to the robot's internal signals, Fig. 2, B (e.g., measuring the amplitude of tool vibrations and correlating these vibrations with the acceleration and speed of the robot in order to optimize its cycle time).
- Evaluation of the robot base vibrations and checking the robot arm alignment for periodic evaluation of its motion accuracy, Fig. 2, C. The vibrations of the base segment are monitored; based on the computed evolution a position alignment check command would be issued. The robot is moved to a point where two sharp marks must align; the verification can be visual or automatic.

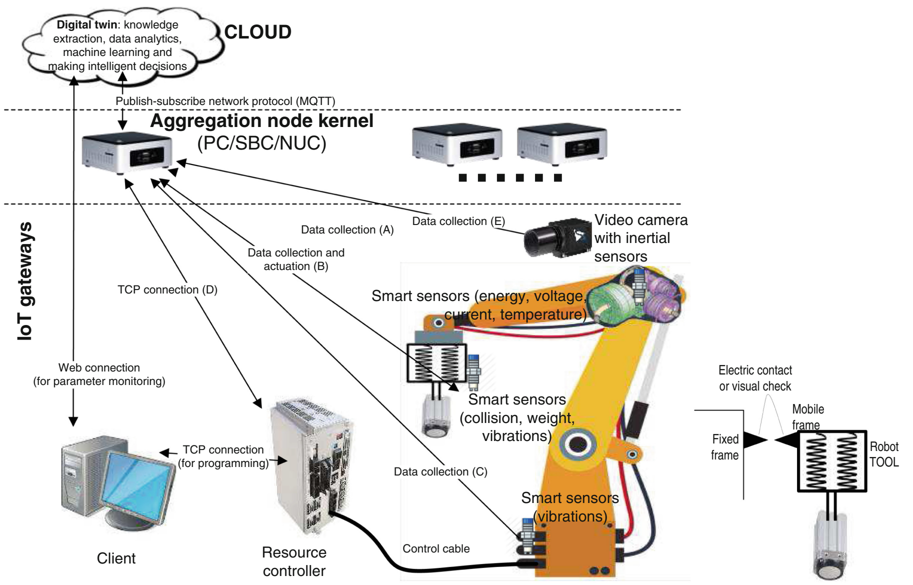

Fig. 2. Interconnections of the robot DT II aggregation node for continuous and operation-based monitoring of QoS and anomaly detection (left); robot alignment device (right)

- Integration of external information used for robot guidance. Variations of the relative position between a fixed camera and a robot can be known by measuring the amplitude of vibrations in the robot base and camera support.

Vibrations relative to a mobile frame (e.g. gripper/tool) are measured using analogue inputs collected relative to the possible movement axis. Vibrations relative to a fixed frame (e.g. robot base - camera support) are measured with inertial sensors. Sensors are connected to the aggregation node which processes data and issues parameter change commands (limit speed/acceleration) or invalidates camera-robot calibration.

4 Experiments with Cloud DT for Adept Robots. Conclusions

Experiments were performed to implement a 4-layer Digital twin framework for Adept industrial robots in a private Cloud IaaS model. Robot data was acquired both from IoT gateways and using the Adept ACE framework that allows data storing into a log file. Adept ACE is based on a series of processes which run in the kernel space of the Adept robot V+ operating system and have access to robot status data.

The data which was collected falls in three categories: (1) *Data characterizing the robot*: AC input [V]; amplifier bus voltage [V]; DC input [V]; amplifier temperature [°C]; encoder temperature [°C]; base board temperature [°C]; duty cycle [% limit]; harmonic drive usage [%]; peak velocity [RPM];peak torque [% max torque]; peak position error [% soft envelope error]; (2) *Data for a robot-serviced belt*: belt velocity [mm/sec]; instance count; instances per minute; active instances; latch faults; (3) *Data characterizing the process*: idle time [%]; processing time [%]; average total time [ms]; targets processed/not processed/per minute; parts processed/not processed/per minute.

The Cygwin Linux tools package was used to exploit the data from the log file; the log file can be processed in real-time by a sequence of tools or a pipeline. The pipeline is executed in a loop based on a configured time interval by the **watch** command. The pipeline consists of three commands: **tail** which extracts the last line from the log file, **cut** which extracts the individual logged values from the line extracted by **tail** (each time when Adept ACE scans the robot status, all values are placed on a single log line and are comma delimited) and sends the output to a custom script **check-and-log** which is based on a configuration file. The pipeline has two functions: i) tests the read values and compares them to a predefined threshold; if the value exceeds the threshold the script can send a notification in the manufacturing cell in order to stop the production or to handle the situation; ii) generates log data for the cloud system. In order to do this the **logger** command is used which generate a standard log message which is sent to the **rsyslog** logging service (Fig. 3).

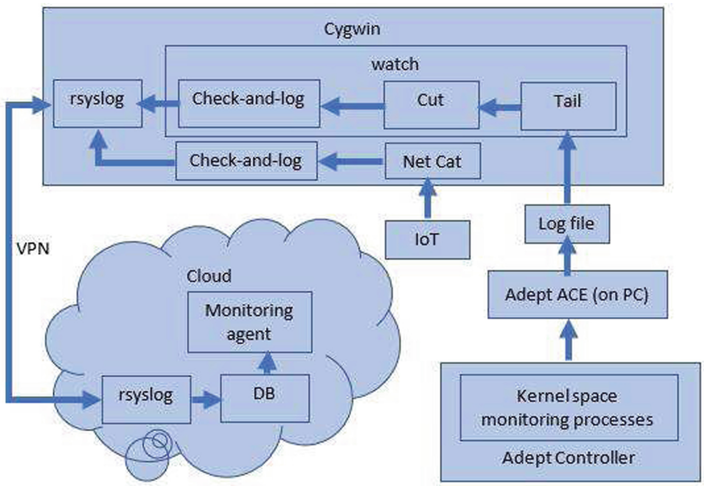

Fig. 3. Processing robot data in cloud DT III and DT IV

The **check-and-log** script is executed in a pipe but is also running standalone in order to obtain other information acquired by the IoT devices, the communication capabilities being offered by Net Cat (a communication tool which can be easily integrated with scripts). When the messages generated by **logger** reach the local **rsyslog** server they get a timestamp (with millisecond resolution) and then are sent over a VPN connection to the cloud where another **rsyslog** server gets the message and stores it into a database (MySQL or Postgres). This remote logging has the advantage that the local storage will not be exhausted and if there are delays in receiving the messages, they will already have an associated timestamp which is not affected by the connection delay [9].

The robot data from the database is analysed by a software agent that uses a heuristic parameter monitoring algorithm; this algorithm can be executed in two ways:

1) Learning a “normal” behaviour of the robot system (learning the variation of the parameters during a robot cycle).
2) Detecting anomalies of the robot based on collected data and trained model.

The first execution mode when the behaviour is learned should not to generate false positive or false negative situations; the model training is restricted on a single “static” robot cycle (which is executed multiple times). A “static” robot cycle means that the cycle is composed by motions using fixed trajectories, constant robot loads, fixed robot speeds and fixed accelerations/decelerations.

For each robot task a model can be trained and used to detect deviations from normal behaviour, based on 3 values defined for each parameter: standard deviation, maximum value and minimum value. During the experiments the digital model of the robot was created based on over 10 K readings. For some parameters such as voltage the model was based on the standard deviation for a robot cycle, in other cases (for example temperatures) the minimum and maximum values have been considered (for the entire robot uptime). Figure 4 presents the data samples acquired for model training.

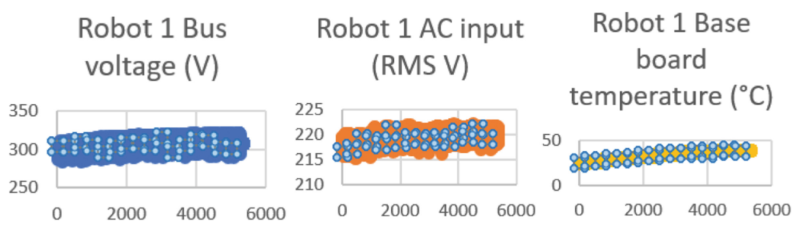

Fig. 4. Robot parameter values used for training

The paper proposes a layered architecture for aggregate digital twins of industrial robots integrated in multi-resource manufacturing systems. These 4-layer digital twins are embedded in individual health monitoring and predictive maintenance programs and collective optimal allocation for product operations at batch level. The robot DT layers span three implementing levels of Cyber-Physical Systems: the edge computing level for data acquisition, the fog computing level for data processing and analysis and the cloud computing level for machine learning and intelligent decision making.

References

1. Morariu, C., Răileanu, S., Borangiu, T., Anton, F.: A distributed approach for machine learning in large scale manufacturing systems. In: Service Orientation in Holonic and Multi-Agent Manufacturing. Studies in Computational Intelligence, vol. 803, pp. 41–52. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-03003-2_3
2. Peres, R.S., Rocha, A.D., Leitao, P., Barata, J.: IDARTS – towards intelligent data analysis and real-time supervision for Industry 4.0. Comput. Ind. **101**, 138–146 (2018). https://doi.org/10.1016/j.compind.2018.07.004
3. Oracle: Digital Twins for IoT Applications: A Comprehensive Approach to Implementing IoT Digital Twins. Redwood Shores (2017)
4. Qia, Q., Taoa, F., Zuoa, Y., Zhaob, D.: Digital twin service towards smart manufacturing. Procedia CIRP **72**, 237–242 (2018). https://doi.org/10.1016/j.procir.2018.03.103
5. Puleri, M., Sabella, R., Osseiran, A.: Cloud robotics. 5G paves the way for mass-market automation. Ericsson Technology Review, 28 June 2016. ISSN 0014-0171
6. Sabella, R., Thuelig, A., Carozza, M.C., Ippolito, M.: Industrial automation enabled by Robotics, Machine Intelligence and 5G. Ericsson Technology Review, February 2018
7. Desai, N.: What is an IoT Gateway and How Do I Keep It Secure? Global Sign Internet GMO Group (2016). https://www.globalsign.com/en/blog/what-is-an-iot-gateway-device
8. Gloria, A., Cercasa, F., Souto, N.: Design and implementation of an IoT gateway to create smart environments. In: Proceedings of the 8th International Conference on Ambient Systems, Networks and Technologies, ANT 2017 (2017). https://doi.org/10.1016/j.procs.2017.05.343
9. Răileanu, S., Anton, F.A., Borangiu, T., Anton, S., Nicolae, M.: A cloud-based manufacturing control system with data integration from multiple autonomous agents. Comput. Ind. **102**, 50–61 (2018)