actuators

MDPI

Article

Modelling, Control Design and Inclusion of Articulated Robots in Cyber-Physical Factories †

Květoslav Belda [1,*], Lukáš Venkrbec [2] and Jan Jirsa [2]

[1] Department of Adaptive Systems, The Czech Academy of Sciences, Institute of Information Theory and Automation, Pod Vodárenskou Věží 4, 18200 Prague 8, Czech Republic
[2] Department of Technical Studies, College of Polytechnics Jihlava, Tolstého 16, 58601 Jihlava, Czech Republic; lukas.venkrbec@vspj.cz (L.V.); jan.jirsa@vspj.cz (J.J.)
* Correspondence: belda@utia.cas.cz

† This paper is an extended version of our paper published in 2023 at the IEEE 27th International Conference on Circuits, Systems, Communications and Computers (CSCC), Rhodes, Greece, 19–22 July 2023; pp. 117–122.

Abstract: This paper addresses the features and limits of the principles and means that provide and support the design of motion control for industrial stationary articulated robots and their involvement in cyber-physical factories as part of the Industry 4.0 concept. The proposed methods are presented herein, from the modelling of kinematics and dynamics considering ideal rigid bodies and principles of classical mechanics, to their application in the design of conventional cascade control and advanced model-based control and use within commercial software tools. The paper demonstrates the modelling principles adapted for control design where a specific novel hierarchical control configuration is outlined. There is an introduction of possible software tools such as Simscape, Robotics Systems Toolbox, RT Toolbox, CIROS and others. It includes the specific aim of the rapid prototyping of robot motion control, which is intended for user development and tuning. In conjunction with conveyor belts, robots-manipulators are essential for cyber-physical factories built on the concept of Industry 4.0. The concept of Industry 4.0 is discussed in respect to the proposed algorithms and software means.

Keywords: industrial robotics; real-time experiment; motion control; Industry 4.0

check for updates

Academic Editor: Zhuming Bi

Received: 17 December 2024
Revised: 21 February 2025
Accepted: 4 March 2025
Published: 6 March 2025

Citation: Belda, K.; Venkrbec, L.; Jirsa, J. Modelling, Control Design and Inclusion of Articulated Robots in Cyber-Physical Factories. *Actuators* **2025**, *14*, 129. https://doi.org/10.3390/act14030129

1. Introduction

Articulated robots belong to a class dominating today's industrial production [1,2]. For this mass use, the control principles in control systems must continuously be subjects for testing, investigation and improvement [3,4]. This follows from a permanent effort to reduce a set of infinitesimal redundant energy inputs that lead to needless additional costs [5,6]. The full robot integration into recent cyber-physical factories (CPF) is a step on the way to the full robotisation of production. Robots are used for handling, as well as for automated manufacturing operations such as welding, painting, machining or assembly in serial and mass production. An example of CPF is shown in Figure 1.

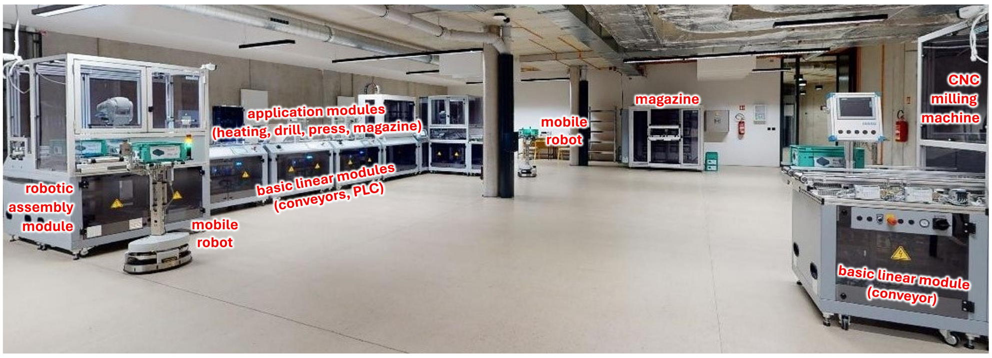


Figure 1. Educational Cyber-Physical Factory by Festo Co. at College of Polytechnic Jihlava.

2. Background of Analysis

There are several investigative perspectives from construction with small applications such as in [7,8], and case studies comparable with industrial applications [9]. The focus is concentrated on problems of system coordination in CPF [10] and the influence of concepts of Industry 4.0 on business operations [11]. The possible future trends are addressed towards collaborative workplaces combining human and smart production entities (machines, manipulators, sensors, etc. [12].

Unsolved research questions (continual challenges) persist in the (i) adjustment of derived ideal mathematical models and their parameters, which need to be in correspondence with the real-world robot features; (ii) which tools are suitable for the hyper prototyping of advanced control design that should reduce energy consumption in future applications; and (iii) how the robots are managed and integrated into sophisticated production procedures and systems represented by CPFs.

This article analyses specific applicable modelling software for improving model-based control design. The gap between theory and software tools used in practice is still large. This is due to the unavailability of theory sourcing and limited development and testing. It is determined by predominately separated small working groups (divisions) without regular interactions between software programmers and developers (experts and scholars) deriving basic principles and practices.

However, the number of robots grows continuously, as is demonstrated in Figure 2, i.e., a constant commissioning of new robots into industrial production every year in spite of fluctuating demand and rising input prices. In this respect, analysis of newly available software tools for industrial use is important for study [13], including the investigation of mutual relations not only within one factory, but connected suppliers using specific organising systems and digital twins in combination with real systems in CPFs [14].

The combination of physical and virtual spaces is referred as cyber-physical systems (CPSs) or CPFs, and the aim of their commissioning is to create a communication interface between the digital and physical worlds by integrating computation, networking and physical assets. It is well-understood that the interconnection between the physical world is represented by hardware (e.g., sensors, actuators, robots) and cyber software (communication, networking and internet) [15]. By using the concepts of Industry 4.0 as virtualisation, communication, smart entities, there is an estimation of the specific possibility to reach a more efficient and optimal production work flow.

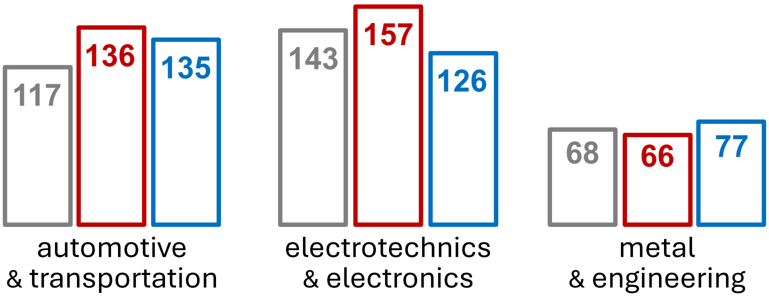


Figure 2. Global industrial robot installations by industry in 2021 (grey), 2022 (red) and 2023 (blue) in thousands [16].

Current modern industrial production is customer-oriented to offer customers a wide-scale portfolio of products with various properties and functions in different variants. It considers specific products such as vehicles-cars in automotive or consumer electronics for households, etc., which is customised according to customer wishes, i.e., the product is tailor-made.

Such a requirement of the market leads to high demands on the organisation and flexibility of the production but also on the timing of forwarding services, suppliers, sellers/traders, as shown in Figure 3. In this situation, Industry 4.0 was applied to pilot production that uses CPF, as shown in Figure 1. CPF, as was mentioned above, represents an approach that combines cyber space (digital twins, computer optimisation, programs, etc.) with the physical world, consisting of real factories with production lines containing real machine tools, robots and other production or manipulation machines and equipment. Both the cyber and real worlds are interconnected via specific envelope/interfaces that enable the producer to provide on-line managing and planning, thus changing and optimising production. This makes it possible to move mass standardised/unified production closer to the a more customer-oriented one.

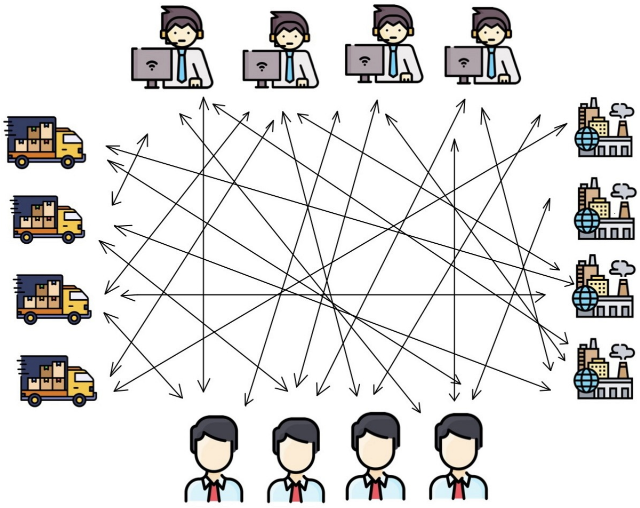


Figure 3. Obsolete concept of piece production (made to order).

The CPF representative depicted in Figure 1 consists of several mutually compatible modules. These modules represent self-autonomous workplaces such as drilling, pressing, heating, assembling, revolving, milling and storage (magazine) modules. They have standardised interfaces (electric and pneumatic power inputs, industrial Ethernet) and Programmable Logic Controllers (PLC), serving as local control units. Such a configuration, together using digital models in software tools RT ToolBox3 Version 2.00A (Mitsubishi) and/or CIROS Studio 7 (Festo), offers the desired production flexibility.

This paper is an extended version of the conference paper [17]. Section 3 deals with the considered representative robot topology. Section 4 deals with the mathematical model synthesis. Then, Section 5 introduces control concepts of both conventional cascade configurations and advanced model predictive procedures in a hierarchical arrangement, combining globally centralised and locally distributed designs [18]. Both concepts are graphically illustrated and compared.

Then, Sections 6 and 7 focus on current software tools and their use in research covering both simulations and the real-run of motion control of common industrial robots. Finally, Section 8 introduces robot integration into intelligent CPFs considering conceptions of Industry 4.0. The paper concludes with a short summary of answers and possible solutions for the indicated open research questions.

3. Considering Robot Topology

Let us consider Mitsubishi RV-4FL-D, an industrial stationary robot, situated in CPF (Figure 4a). It is considered for model synthesis, simulations and real experiments [19].

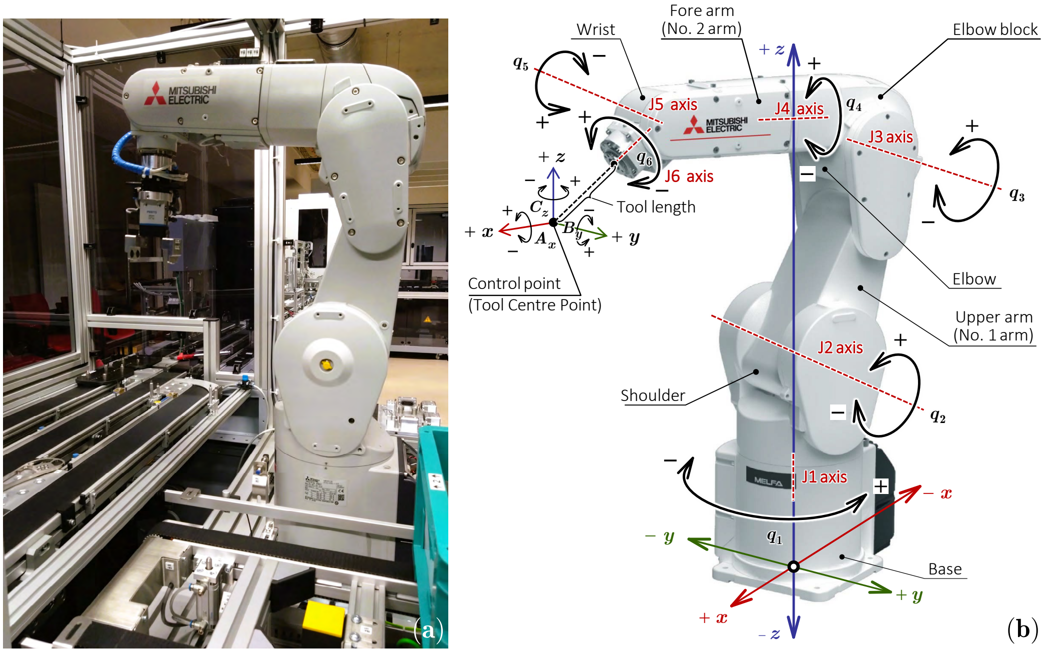


Figure 4. Robot RV-4FL-D (**a**) in CPF in assembly module and (**b**) with indicated coordinates [19].

The topology of the robot has 6 degrees of freedom (DOF). The robot consists of a base and 6 arms (shoulder, upper arm, elbow, elbow block, fore arm and wrist with flange) up to the robot's flange. It enables the user to set and control the Cartesian coordinates x, y and z for positions and A_x, B_y and C_z for the orientation of the robot tool centre point (TCP), set and control independent joint coordinates, or drive coordinates $q_{1–6}$.

4. Mathematical Model Synthesis

This section shows the synthesis of the mathematical model for simulation and as well as for the model-based control design. Ideally, rigid bodies are considered. This includes the synthesis of both forward and inverse problems for the robot kinematics and dynamics for the robot in Figure 4b. The obtained models of kinematics serve as trajectory planning, especially for the conversion of user TCP coordinates into joint coordinates, which is expected on robot drives.

4.1. Model of Robot Kinematics

To obtain the kinematic model, a Denavit–Hartenberg (DH) concept can be used. The concept defines the transformations between the joint coordinate space (drive space) and the operational coordinate space (tool centre point (TCP), robot end effector) in both directions: the forward direction ($q \rightarrow y$, joint/drive coordinates into TCP Cartesian coordinates) and the inverse direction ($y \rightarrow q$, TCP into joint/drive coordinates), where joint and operational coordinates q and y are defined for the considered robot as follows: [1]

$$
\begin{aligned}
q &= \left[q_1,\, q_2,\, q_3,\, q_4,\, q_5,\, q_6 \right]^T, \\
y &= \left[x,\, y,\, z,\, A_x,\, B_y,\, C_z \right]^T.
\end{aligned}
\tag{1}
$$

4.1.1. Forward Problem

For the 6 DOF robot, the forward problem represents a specific combination of three basic motions along axes x, y, and z, and three basic rotations around these three axes. The motions and rotations are parameters involved in the respective complex matrices. These matrices represent the individual transformations between the robot joints, i.e., they are connected to the corresponding individual robot bodies, as follows:

$$
\begin{aligned}
F = T_0^1\ T_1^2(q_1)\ T_2^3(-q_2)\ T_3^{\bar{3}}(-q_3)\ T_{\bar{3}}^4(90^\circ), \\
T_4^5(q_4)\ T_5^6(q_5)\ T_6^F(q_6)\ r_0,
\end{aligned}
\tag{2}
$$

where $F = \left[x_F,\, y_F,\, z_F,\, 1 \right]^T$ represents considered TCP and $r_0 = \left[0,\, 0,\, 0,\, 1 \right]^T$ represents the general origin for coordinate systems.

In (2), the first line represents TCP positioning and the second line orientation in TCP, i.e., the orientation of tool or gripper principal axis. It corresponds to a pattern from nature, as do the names in Figure 4b and Table 1, which are taken from the biology of the human arm.

Table 1. D-H Parameters.

Robot	**D-H Parameters**			
Body Names	$\boldsymbol{q_i}$	$\boldsymbol{d_i}$	$\boldsymbol{a_i}$	$\boldsymbol{\alpha_i}$
Base (0–1)	0°	h_0	0	0°
Shoulder (1–2)	q_1	h_1	0	90°
Upper arm (2–3)	$90^\circ - q_2$	0	ℓ_2	0°
Elbow ver. (3–$\bar{3}$)	$-q_3$	0	r_3	90°
Elbow hor. ($\bar{3}$–4)	90°	ℓ_3	0	0°
Fore arm (4–5)	$-90^\circ - q_4$	ℓ_4	0	-90°
Wrist (5–6)	q_5	ℓ_5	0	90°
Flange (6–T)	q_6	ℓ_6	0	0°

The individual motions and rotations, and their combinations mentioned above, are realised by transformation matrices expressed generally by one variable complex matrix, as follows:

$$
\begin{bmatrix}
\cos q_i & -\cos \alpha_i\ \sin q_i & \sin \alpha_i\ \sin q_i & a_i\ \cos q_i \\
\sin q_i & \cos \alpha_i\ \cos q_i & -\sin \alpha_i\ \cos q_i & a_i\ \sin q_i \\
0 & \sin \alpha_i & \cos \alpha_i & d_i \\
0 & 0 & 0 & 1
\end{bmatrix}
\tag{3}
$$

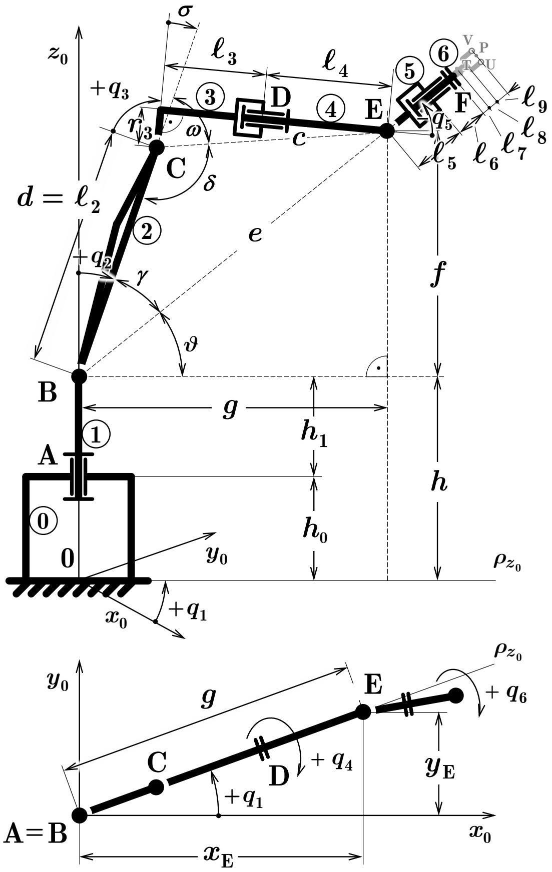


Figure 5. Two-dimensional views for robot kinematic analysis [17]. Individual numbers ⓪–⑥ represent numbers of robot bodies, see Figure 4b.

The matrices contain constant and/or variable parameters q_i, d_i, a_i and α_i—so called D-H parameters relating to the kinematic chain of the robot [1]. Their values are listed in Table 1. The number of variable rotations q_i is 6 according to DOF of the considered robot. Other indicated constants h_0, h_1, ℓ_2, r_3, ℓ_3, ℓ_4, ℓ_5 and ℓ_{6-9} include vertical distances, axis offsets and lengths of robot arms. Moreover, the joint between 'Elbow ver. (= Elbow vertical)' and 'Elbow hor. (= Elbow horizontal)' is fixed as an immovable right-angle joint $\bar{3}$.

4.1.2. Inverse Problem

The inverse problem can be solved either numerically as a solution of the transformations expressed by (2) or analytically. Here, the analytical solution is demonstrated using geometrical relations depicted in Figure 5. The following expressions introduce the solution.

$$q_1 = \operatorname{atan2}(y_E, x_E), \tag{4}$$

$$q_2 = \frac{\pi}{2} - (\gamma - \vartheta), \tag{5}$$

$$q_3 = \frac{3}{2}\pi - \beta - \gamma, \tag{6}$$

$$f = z_E - h = z_E - (h_0 + h_1), \quad g = \sqrt{x_E^2 + y_E^2}, \quad e = \sqrt{f^2 + g^2}, \tag{7}$$

$$c = \sqrt{r_3^2 + (\ell_3 + \ell_4)^2}, \tag{8}$$

$$\gamma = \arccos\frac{d^2 + e^2 - c^2}{2\,d\,e}, \quad \delta = \arccos\frac{c^2 + d^2 - e^2}{2\,c\,d}, \tag{9}$$

$$\vartheta = \operatorname{atan2}(f, g), \quad \omega = \operatorname{atan2}(\ell_3 + \ell_4, r_3). \tag{10}$$

To obtain the coordinates of point E for specific cases, let point E be determined using appropriate transformation chains based on orientation angles Ax, By, Cz from end points, specifically F (Flange) or tool points T, U, V and P—see [20].

4.2. Model of Robot Dynamics

The model of the robot dynamics is derived using Lagrange equations of the second type:

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{q}}\right)^T - \left(\frac{\partial \mathcal{L}}{\partial q}\right)^T = \xi, \tag{11}$$

$$\mathcal{L} = V(q, \dot{q}) - P(q), \tag{12}$$

$$\text{i.e.,}\quad \frac{d}{dt}\left(\frac{\partial V}{\partial \dot{q}}\right)^T - \left(\frac{\partial V}{\partial q}\right)^T + \left(\frac{\partial P}{\partial q}\right)^T = \xi, \tag{13}$$

where ξ are generalised force effects and V is kinematic energy as follows:

$$V = \sum_{i=1}^{n} V_i = \frac{1}{2}\sum_{i=1}^{n}\left(v_i^T\, m_i\, v_i + \omega_i^T\, \mathcal{I}_i\, \omega_i\right). \tag{14}$$

The kinetic energy V in (14) for translation and rotation along the principal axes can be simply expressed as follows:

$$V = \frac{1}{2}\sum_{i=1}^{n}\left(m_i\, v_i^2 + I_i\, \omega_i^2\right), \tag{15}$$

where n is a number of robot bodies and quantities m_i, I_i, v_i and ω_i are weights, moments of inertia, translation and angular speeds, respectively. P is potential energy:

$$P = \sum_{i=1}^{n} P_i = \sum_{i=1}^{n} m_i\, g\, z_i, \tag{16}$$

where quantities z_i represent the appropriate vertical distances of the centres of gravity of individual bodies and g is gravitational acceleration. Expressing the terms of (13) leads to the dynamic equations of motion as follows (inverse dynamics) [1]:

$$M(q, \dot{q})\, \ddot{q} + N(q, \dot{q})\, \dot{q} + g(q) = \tau, \tag{17}$$

or (forward dynamics):

$$\ddot{q} = -M^{-1} N\, \dot{q} - M^{-1} g + M^{-1} \tau, \tag{18}$$

$$\ddot{q} = -M^{-1} N\, \dot{q} + u, \tag{19}$$

where modification $u = -M^{-1}g + M^{-1}\tau$ removes the effects of gravitation from the model and simplifies control design. The expression (18) can be performed in continuous or discrete state-space forms

$$\left.\begin{array}{lcl} \dot{x} = A(x)\, x + B\, u & \rightarrow & x_{k+1} = A(x_k)\, x_k + u_k \\ y = C\, x & \rightarrow & y_k = C\, x_k \\ \tau = M\, u + g & \rightarrow & \tau_k = M\, u_k + g \end{array}\right\} \tag{20}$$

The equations in (20) are prepared for simulation and control design.

5. Control Concepts and Principles

This section introduces specific representatives of non-model and model-based control approaches used for applications in industrial stationary (not only articulated) robots. The first one presents a common approach used in industry without any optimisation for future outlook, but is proven. The second represents a strategy enabling the future development and inclusion of additional user requirements as specific constraints on motion increments, smoothing for specific horizons, and energy optimisation, i.e., better energy input distribution in multi-body systems for robots. Note that the success and accuracy of the robot motion depends on the appropriately designed reference trajectory, considering technological and geometric constraints as well.

5.1. Conventional Cascade Control

Let us consider continuous and discrete PID controllers as representatives of the conventional non-model-based control approach. Their usual basic forms are as follows.

- continuous form:

$$u(t) = K_p \left(e(t) + \frac{1}{T_i} \int_0^t e(t)\,\mathrm{d}t + T_d\,\dot{e}(t) \right), \tag{21}$$

- discrete incremental form:

$$u_k = u_{k-1} + K_p \left((e_k - e_{k-1}) + \frac{1}{T_i}\, e_k\, T_s + T_d\, \frac{e_k - 2\,e_{k-1} + e_{k-2}}{T_s} \right). \tag{22}$$

The form (22), if applied to the drive control, i.e., the actuation of individual robot joints, is configured to the cascade configuration [21]. It consists of three interrelated loops: positional, speed and current, see Figure 6.

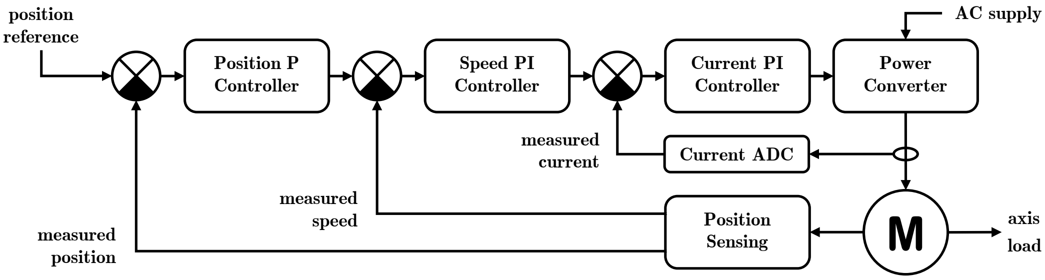


Figure 6. Conventional PID-based cascade control for individual joints/axes/drives [17,21,22].

The mentioned concept represents still the independent (distributed, local) control approach from the robot point of view. All internal interactions given by a robot dynamic behaviour are considered as external disturbances.

5.2. Advanced Model Predictive Control

The discrete model predictive control is the representative of a model-based control with a specific optimality criterion:

$$\min_{\Delta U_k} J_k\,(\hat{Y}_{k+1},\, W_{k+1},\, \Delta\, U_k), \tag{23}$$

subject to the state-space model (20) and constraints:

$y_{min} \leq y_{j+1} \leq y_{max}$ and $u_{min} \leq u_j \leq u_{max}$, where $\hat{Y}_{k+1}$, W_{k+1} and $\Delta\, U_k$ are vectors of output predictions, reference values and increments of searched control actions

$u_k = u_{k-1} + \Delta\, u_k$, respectively; $j = k, \cdots, k+N$ and N is a prediction horizon. The cost of the criterion (23) is usually expressed by quadratic function:

$$J_k = ||Q_{YW}\,(\hat{Y}_{k+1} - W_{k+1})||_2^2 + ||Q_{\Delta\, U}\, \Delta\, U_k||_2^2. \tag{24}$$

Figure 7 shows the novel concept of the application of predictive control. A global slower part has online optimisation [23] and a local fast part uses pre-computed control laws [18]. The concept uses the same computation principle.

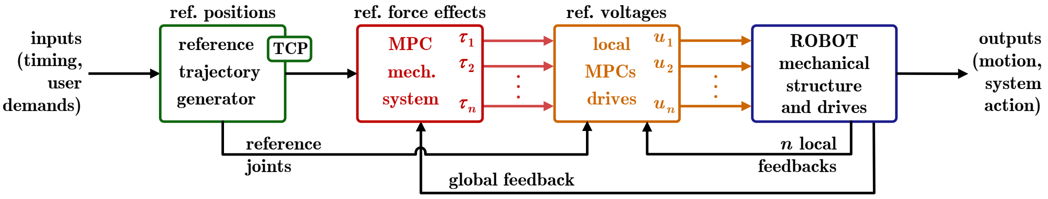


Figure 7. Novel model predictive control concept for robot dynamics and drives [17,18,23].

6. Software Tools

To provide adequate robot behaviour analysis and control design, it is necessary to use suitable software tools. These support users to obtain detailed robot models and simulate them, and to realise real-run control and data acquisition from the physical robot motion. Outside of the mentioned tools, other modelling software such as NX or Inventor and others should be considered. All these tools serve the determination of realistic (ideal) physical parameters such as moments of inertia, weights, lengths and positions of centres of gravity, which all are important for the construction of an adequate mathematical model.

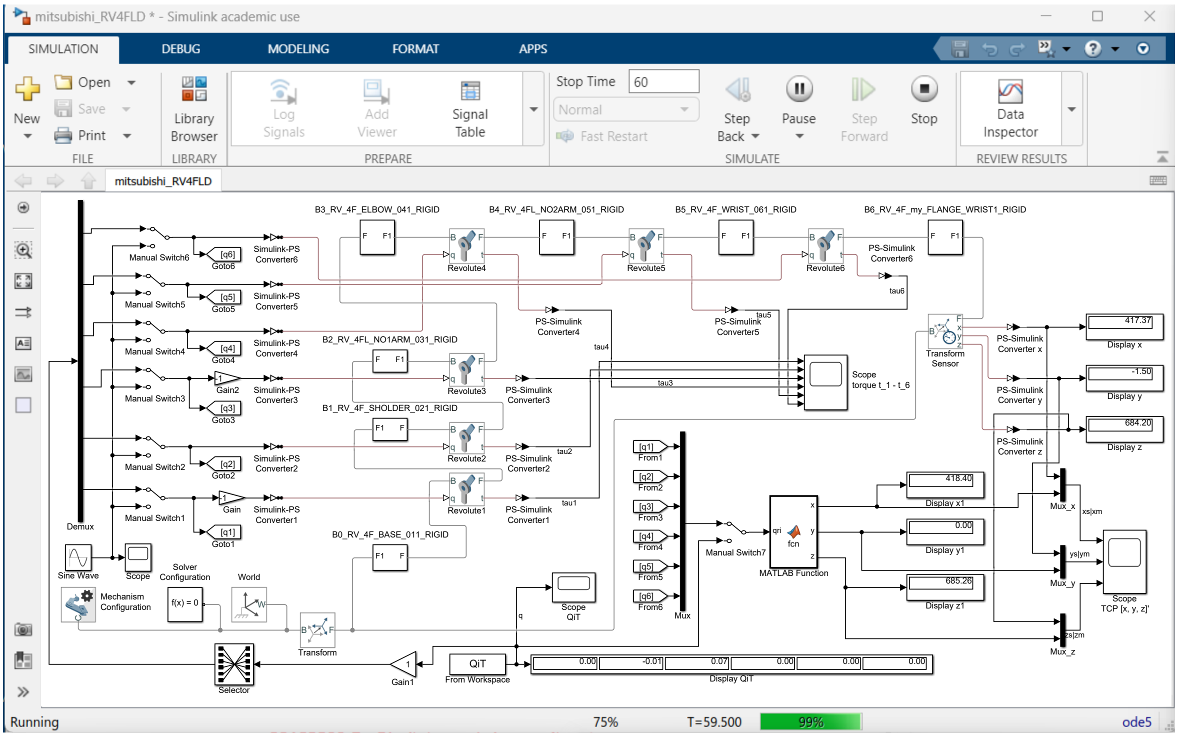
6.1. Simscape Multibody Lib—MathWorks

Simscape Libraries, specifically Simscape Multibody library [24] included in Simulink, serves to verify derived models and 3D CAD models, especially for rapid drive prototyping in the construction phase, when only the CAD model is available, but there are no equations of motion (model of robot dynamics). In this paper, the Multibody library was used for simulations of dynamic robot behaviour, see Figure 8: Simulink model based on blocks of Multibody library, and Figure 9: 'Mechanics Explorer' involving imported 3D CAD model, and keeping real physical parameters from used original CAD applications (NX or Inventor) as mentioned above.

6.2. Robotics System Toolbox—MathWorks

Similarly to the previous section, the plain MATLAB R2024b environment with the Robotics System Toolbox [25] is suitable for simulations with a rich function set as well. It enables users to have all procedures in hand, i.e., from trajectory planning, to control design and its verification. Toolbox is specifically suitable for the verification of kinematic transformations and collision tests for user defined trajectories. A creation of the structure describing the robot is shown in Figure 10 and the listing of the created structure in the MATLAB command line window is in Figure 11.

The model defined and used in this toolbox contains all physical parameters such as arm lengths, masses, moments of inertia, centres of gravity and geometrical envelopes of robot bodies. The motion during a simulation can be visualised, including collision analysis and motion trajectory display (Figure 12). The wireframe points were calculated using the kinematics equations in Section 4.1, as well as the figures below. They show simulated joint and TCP coordinates. In this paper, the MATLAB software environment including Simulink and its toolboxes and libraries was used due to the possible importation of a 3D CAD robot model [26]. Furthermore, for the implementation of derived models and for the processing of measured data by RT Toolbox (Mitsubishi), see the figures in the next section.



Stop Time 60

Running

Figure 8. Simulink model of Mitsubishi robot RV-4FL-D using Simscape Multibody Library [3,17].

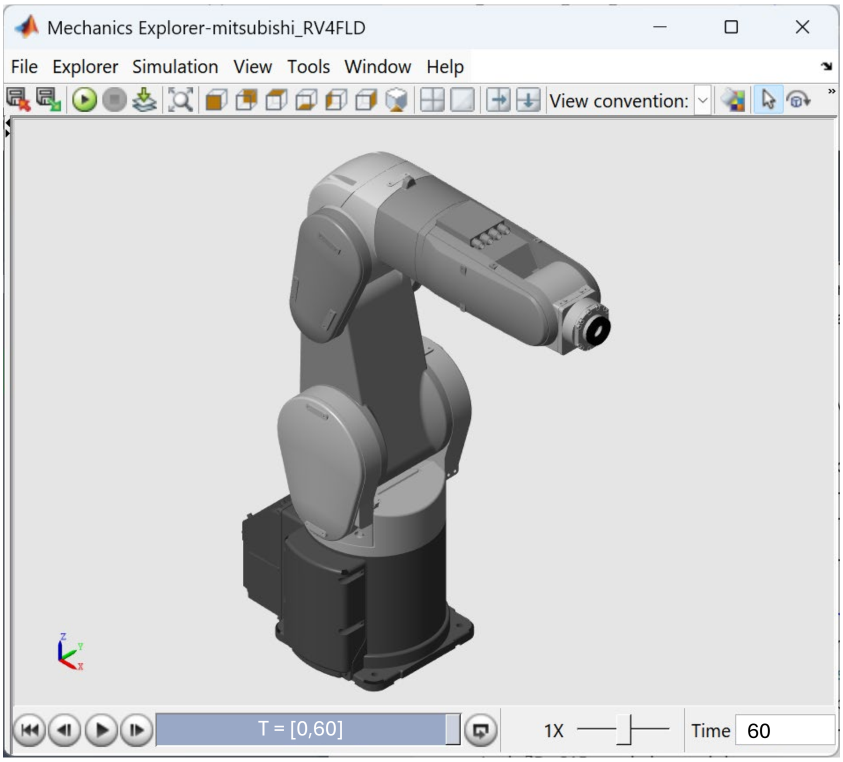


Figure 9. Mechanics Explorer with 3D model of Mitsubishi robot RV-4FL-D [3].

```
robot = rigidBodyTree; % units in *.stl files [mm]
frame = rigidBody('frame');
addVisual(frame,"Mesh",'B00_RV4FL_BASE.stl');
jntf = rigidBodyJoint('jntf','fixed');
tform = trvec2tform([0 0 0]);
setFixedTransform(jntf,tform); frame.Joint = jntf;
addBody(robot,frame,'base')
arm1 = rigidBody('arm1');
addVisual(arm1,"Mesh",'B01_RV4FL_SHOLDER.stl');
jnt1 = rigidBodyJoint('jnt1','revolute');
Jnt1.HomePosition = 0; jnt1.JointAxis = [0 0 1];
tform = trvec2tform([0 0 193]); fi1 = 0;
tform(1:3,1:3) = [cosd(fi1) -sind(fi1) 0; ...
sind(fi1) cosd(fi1) 0; 0 0 1]; %rotz(0)
tform = tform*trvec2tform([0 0 0]);
setFixedTransform(jnt1,tform); arm1.Joint = jnt1;
addBody(robot,arm1,'frame')
arm2 = rigidBody('arm2');
addVisual(arm2,"Mesh",'B02_RV4FL_NO1ARM.stl');
jnt2 = rigidBodyJoint('jnt2','revolute');
Jnt2.HomePosition = 0; jnt2.JointAxis = [0 -1 0];
tform = trvec2tform([0 0 157]);  fi2 = 0;
tform(1:3,1:3) = [cosd(fi2) 0 sind(fi2); ...
0 1 0; -sind(fi2) 0 cosd(fi2)]; %roty(0)
tform = tform*trvec2tform([0 0 0]);
setFixedTransform(jnt2,tform); arm2.Joint = jnt2;
addBody(robot,arm2,'arm1')

                ⋮
arm6 = rigidBody('arm6');
addVisual(arm6,"Mesh",'B06_RV4FL_FLANGE.stl');
jnt6 = rigidBodyJoint('jnt6','revolute');
Jnt6.HomePosition = 0; jnt6.JointAxis = [1 0 0];
tform = trvec2tform([85 0 0]); fi6 = 0;
tform(1:3,1:3) = [1 0 0; 0 cosd(fi6) -sind(fi6); ...
0 sind(fi6) cosd(fi6)]; %rotx(0)
tform = tform*trvec2tform([0 0 0]);
setFixedTransform(jnt6,tform); arm6.Joint = jnt6;
addBody(robot,arm6,'arm5')
```

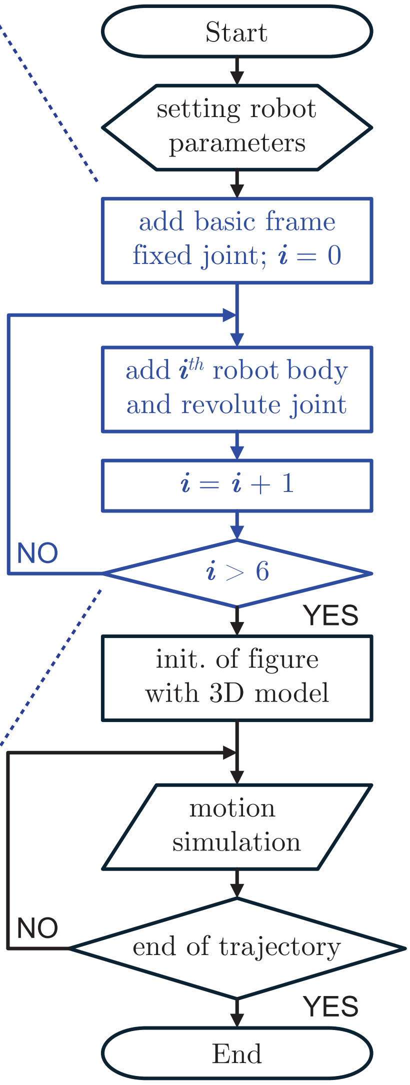


Figure 10. Creation of the structure 'robot' as 'rigidBodyTree' in Robotics System Toolbox. (Real configuration for considered Mitsubishi robot RV-4FL-D).

```
>> showdetails(robot)
--------------------
Robot: (7 bodies)
Idx    Body Name    Joint Name    Joint Type    Parent Name(Idx)   Children Name(s)
---    ---------    ----------    ----------    ----------------   ----------------
1        frame          jntf         fixed             base(0)   arm1(2)
2         arm1          jnt1      revolute            frame(1)   arm2(3)
3         arm2          jnt2      revolute             arm1(2)   arm3(4)
4         arm3          jnt3      revolute             arm2(3)   arm4(5)
5         arm4          jnt4      revolute             arm3(4)   arm5(6)
6         arm5          jnt5      revolute             arm4(5)   arm6(7)
7         arm6          jnt6      revolute             arm5(6)
--------------------
```

Figure 11. Structure 'robot' in Robotics System Toolbox. (Real configuration for considered Mitsubishi robot RV-4FL-D listed in the MATLAB command line).

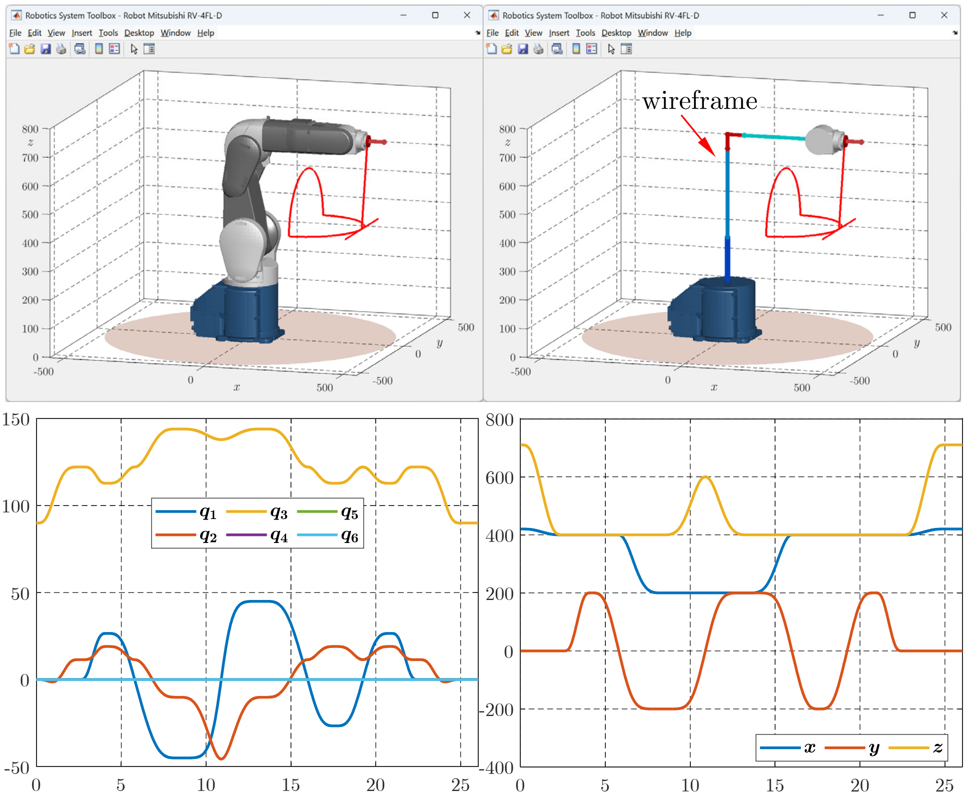
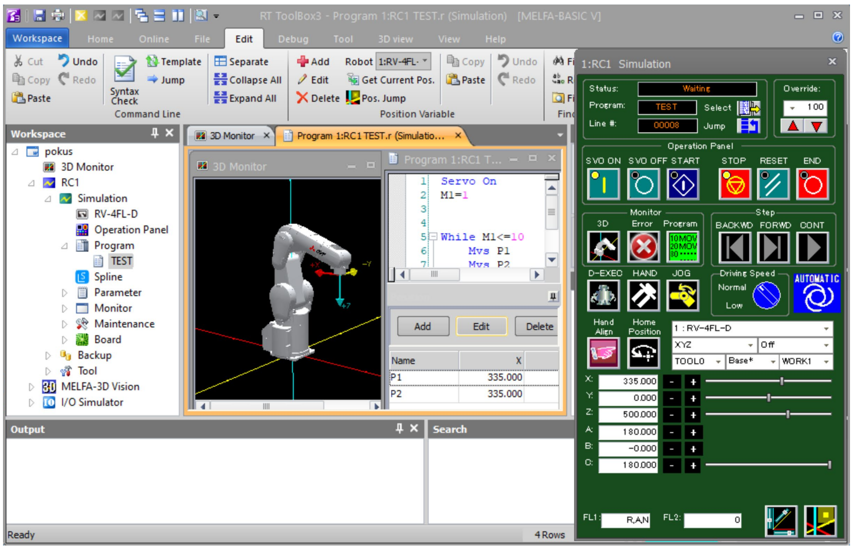


Figure 12. MATLAB Figure with 'rigidBodyTree' robot class RV-4FL-D; and time behaviours of joint and TCP coordinates.

6.3. RT Toolbox—Mitsubishi Electric

RT Toolbox represents a suitable software tool that enables users to program robot motion and to verify it, including 3D visualisation and executing the real control of the specific class of industrial robots, including data acquisition options [27]. The data acquisition depends on the type of control system of the Mitsubishi robot.

In this paper, this RT Toolbox was used for the verification of the designed motion trajectory and for real experiments on the described robot Mitsubishi RV-4FL-D as shown in Figure 13. For measured data of real experimental runs tracking testing trajectory see Section 7.



```
Servo On
M1=1


While M1<=10
    Mvs P1
    Mvs P2
```

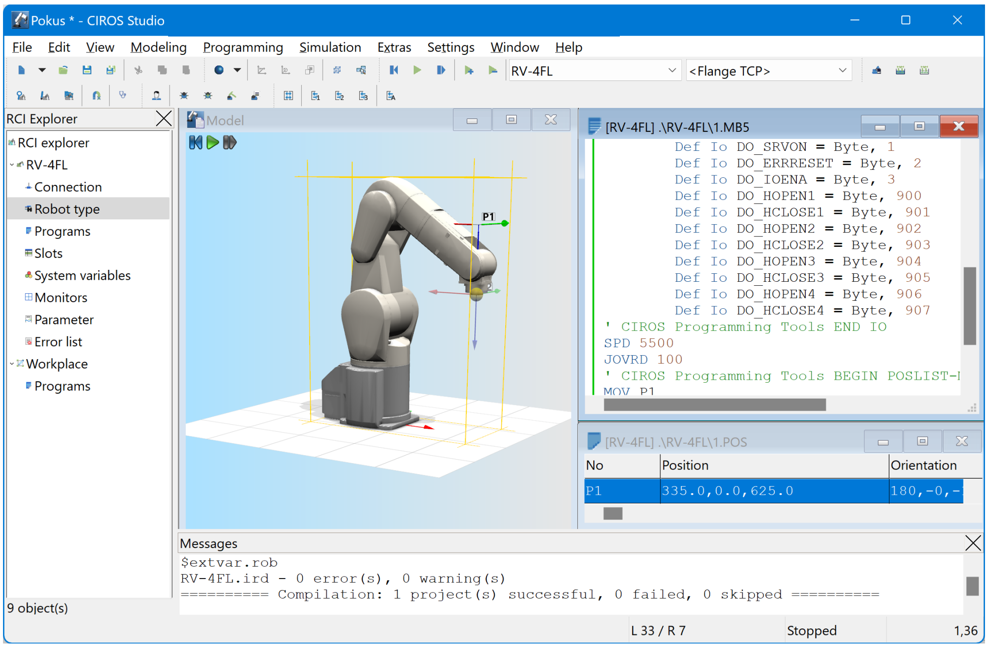
Name	X
P1	335.000
P2	335.000

Status: Waiting
Program: TEST
Line #: 00008
Override: 100
1 : RV-4FL-D
XYZ
Off
TOOL0
Base*
WORK1
X: 335.000
Y: 0.000
Z: 500.000
A: 180.000
B: -0.000
C: 180.000
FL1: R,A,N
FL2: 0

Figure 13. RT Toolbox environment with robot RV-4FL-D.

6.4. CIROS Studio—Festo

CIROS Studio is an industrially oriented software tool, containing analogous features such as RT Toolbox. It includes an environment for creating 3D simulation models and for programming with wide debugging and simulation options, see Figure 14. The data of robot statements or input commands can be managed and changed using industrial interfaces, and user-programmed with approaches via Python 3.13.1.



```
            Def Io DO_SRVON = Byte, 1
            Def Io DO_ERRRESET = Byte, 2
            Def Io DO_IOENA = Byte, 3
            Def Io DO_HOPEN1 = Byte, 900
            Def Io DO_HCLOSE1 = Byte, 901
            Def Io DO_HOPEN2 = Byte, 902
            Def Io DO_HCLOSE2 = Byte, 903
            Def Io DO_HOPEN3 = Byte, 904
            Def Io DO_HCLOSE3 = Byte, 905
            Def Io DO_HOPEN4 = Byte, 906
            Def Io DO_HCLOSE4 = Byte, 907
' CIROS Programming Tools END IO
SPD 5500
JOVRD 100
' CIROS Programming Tools BEGIN POSLIST-
MOV P1
```

No	Position	Orientation
P1	335.0,0.0,625.0	180,-0,-

```
$extvar.rob
RV-4FL.ird - 0 error(s), 0 warning(s)
========== Compilation: 1 project(s) successful, 0 failed, 0 skipped ==========
```

Figure 14. CIROS Studio environment with robot RV-4FL-D.

In this paper, CIROS Studio was considered powerful alternative to RT Toolbox for the verification of motion trajectory and for real experiments on considered robot RV-4FL-D.

7. Simulation and Real Experiment

Modelling includes both mathematical and 3D modelling, and the planning of motion trajectories, i.e., it represents a number of complex procedures. This section show results in the form of time histories of simulated and measured physical quantities of robot motion.

7.1. Simulation

Specifically, let us first consider the definition of reference motion in Figure 15 or its equivalent code shown in Figure 16. It shows the universal description of the motion geometry in G-code for motion trajectory generation, i.e., path interpolation and time parameterisation. Using listed G-code via MATLAB user functions, the simulation runs were realised. Figure 17a–c show simulated input data processed by a robot control system intended for real experiment. A record of the simulation is shown in Figure 12 that proves the correctness of user-designed trajectory.

7.2. Real Experimental Results

For the code in Figure 16, Figure 18a–c shows measured state quantities using RT Toolbox specifically as follows: angular position, angular speed, current (torque currents) for individual joints—appropriate drives along defined trajectory, i.e., each containing 6 time histories corresponding to 6 driven joints/axes. The presented time histories were captured by RT Toolbox for sampling period $T_s = 7.1\,\text{ms}$ [27]. Current profiles indicate load from motion and the robot arm itself from joint 1 to joint 6. The biggest load has to be covered by joint 2, which produces the main motion of the robot access system (the first three axes). On the other side, the orientation system (wrist, the last three axes) represents only a particular motion. It was reduced by the reference trajectory. Figure 18b depict x, y and z Cartesian coordinates as individual time histories of resultant 3D trajectory.

```
N010  G19
N020  G00  X400  Y0     Z400
N030  G01  X400  Y200   Z400
N040  G01  X400  Y0     Z400
N050  G02  X200  Y-200  Z400  I-200  J0  K0
N060  G02  X200  Y200   Z400  I0   J200  K0
N070  G02  X400  Y0     Z400  I0  J-200  K0
N080  G01  X400  Y-200  Z400
N090  G01  X400  Y200   Z400
N100  G01  X400  Y0     Z400
N110  G00  X420  Y0     Z710
```

```
 % HOME position of the robot RV-4FL-D:
 % Joint Jog mode [deg]:
 % J1 = 0;  J2 = 0;  J3 = 90;
 % J4 = 0;  J5 = 0;  J6 =  0;
 % XYZ  Jog mode   [mm]:
 %  X = 420;   Y =  0;  Z = 710;
 % Ax = -180; By = 90; Cz = 180;
```

Figure 15. G-code of testing trajectory in mm.

```
N010  Servo On
N020  M1=1
N030  While M1<=10
N040  Mvs  P1
N050  Mvs  P2
N060  Mvr  P2, Pr23, P3
N070  Mvr  P1,  Pr2, P4
N080  Mvr  P4,  Pr3, P5
N090  Mvr  P5,  Pr1, P6
N100  Mvs  P7
N110  Mvs  P8
N120  Mvs  P9
N130  Mvs P10
N140  M1 = M1 + 1
N150  WEnd
N140  Hlt
```

```
Definition of used points Pi in mm:
Pi  =(   x,    y,    z,  Ax,  By,  Cz)(Fl1,Fl2)
P1  =(+400,   +0,+400,+180,  +0,+180)( 7 , 0 )
P2  =(+400,+200,+400,+180,  +0,+180)( 7 , 0 )
Pr23=(+400,+100,+400,+180,  +0,+180)( 7 , 0 )
P3  =(+400,   +0,+400,+180,  +0,+180)( 7 , 0 )
Pr2 =(+341,-141,+400,+180,  +0,+180)( 7 , 0 )
P4  =(+200,-200,+400,+180,  +0,+180)( 7 , 0 )
Pr3 =(+200,   +0,+600,+180,  +0,+180)( 7 , 0 )
P5  =(+200,+200,+400,+180,  +0,+180)( 7 , 0 )
Pr1 =(+341,+141,+400,+180,  +0,+180)( 7 , 0 )
P6  =(+400,   +0,+400,+180,  +0,+180)( 7 , 0 )
P7  =(+400,-200,+400,+180,  +0,+180)( 7 , 0 )
P8  =(+400,+200,+400,+180,  +0,+180)( 7 , 0 )
P9  =(+400,   +0,+400,+180,  +0,+180)( 7 , 0 )
P10 =(+420,   +0,+710,+180,  +0,+180)( 7 , 0 )
P0  =(+420,   +0,+710,+180,  +0,+180)( 7 , 0 )
```

`Note:` $\underline{341} = 200 + 141 (= 200 \sin 45^\circ)$ and $\underline{141} = 200 \sin 45^\circ$

Figure 16. Code (Melfa Language for RT Toolbox).

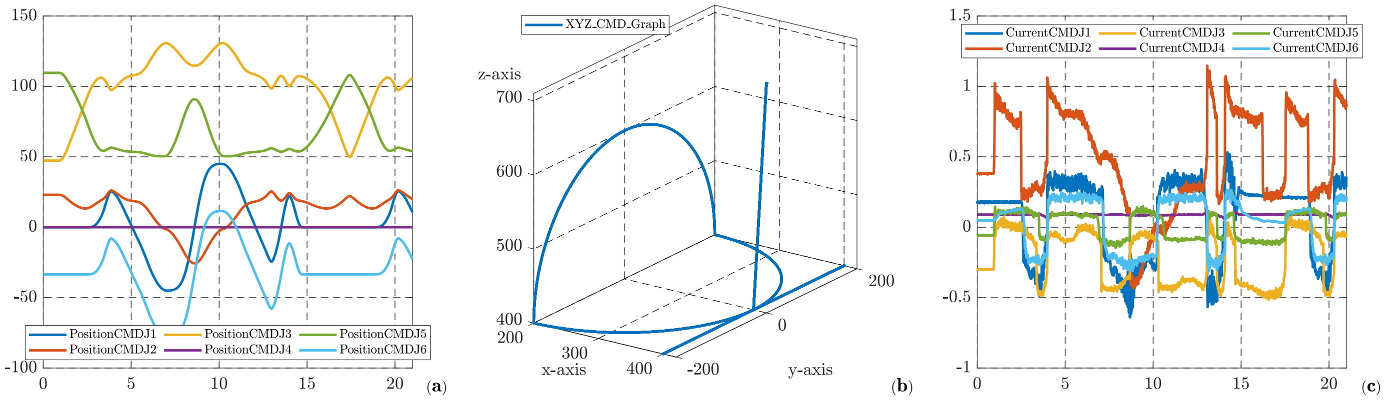


Figure 17. (**a**) joint command 1–6 (deg); (**b**) reference XYZ graph (mm); (**c**) current command 1–6 (Arms).

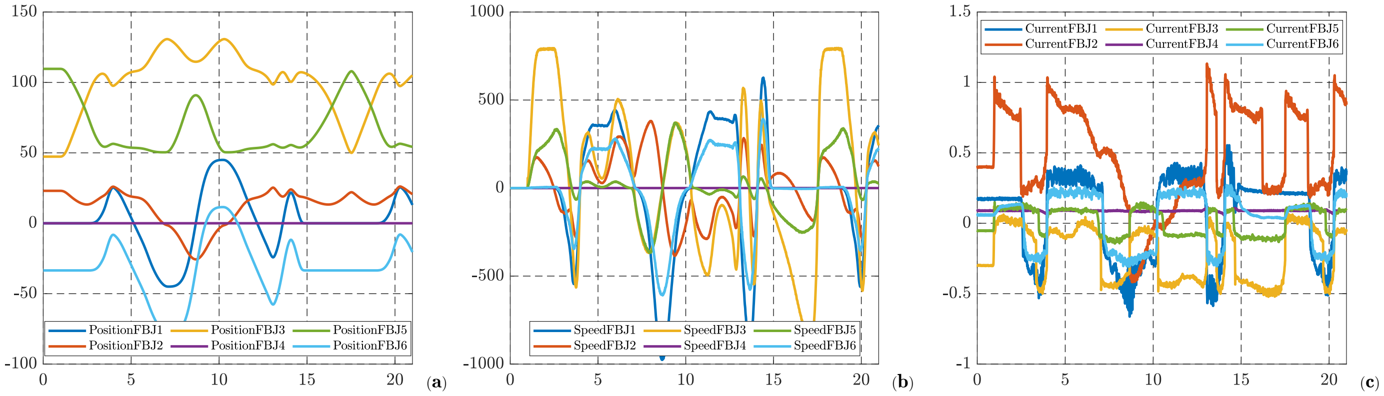


Figure 18. (**a**) joint feedback 1–6 (deg); (**b**) speed feedback 1–6 (deg/s); (**c**) current feedback 1–6 (A) [17].

8. Robot Integration in Cyber-Physical Factories

In the previous sections, the creation of digital models was introduced. If such models, robots or other devices should be integrated into the complex production process, then such a task represents a totally different point of view. A basic element for the integration of the robot(s) (or any other unit(s)) is a digital interface providing a standardised communication between units and the outside world. However, the interface itself is not sufficient for the full integration of the robot into the production process, since full integration means both the ability to offer and request individual operations or steps of the production process as well as the ability to dynamically react to events such as error states, adding a new unit (element), changing the parameters of the production process, changing customer requirements, etc.

For such a level of integration, it is necessary to provide the surrounding system with comprehensive information about the capabilities, parameters and status of the given device (robot). This consideration leads directly to the concept of Industry 4.0 (I4.0) components, as defined in [28] and the concept of a real digital twin (full digital twin), Figure 19.

According to the I4.0 definition of a component [28], a given object becomes a component if it is equipped with a so-called administrative shell. That is, a digital envelope that contains all the above-mentioned parameters, states and information about the physical object together with a unified interface for communication with the environment. The mutual communication of the components created in this way takes place exclusively through these administrative shells (see Figure 20). The data structure of this envelope itself is not part of the definition of the I4.0 component, but the Asset Administration Shell (ASS) standard, defined in [29,30], appears to be a suitable data model for this case.

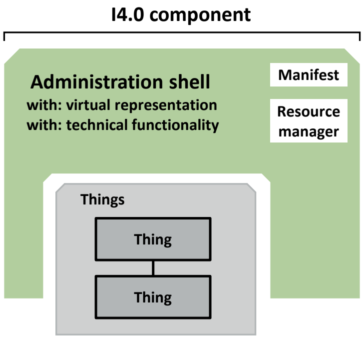


Figure 19. Principle of I4.0 Component [28].

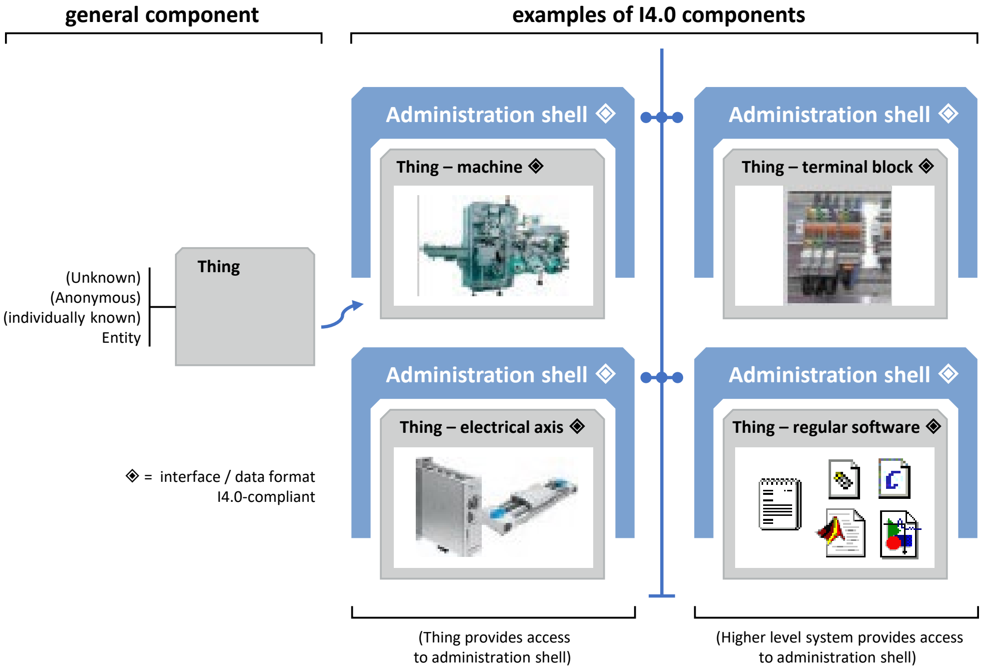


Figure 20. Set of I4.0 components and their common communication.

8.1. Asset Administration Shell

The specification of ASS defines common form of digital component envelope I4.0 and it is originated and still developed by the international organisations Platform Industrie 4.0, VDE/VDI and ZVEI [31]. Specifications [29,30] contain detail description of data model, design way, class description and communication ways in envelope level. The digital envelope in ASS point of view can be split in passive and active parts, where passive part is mandatory:

- passive part contains descriptions of parameters, properties and object abilities in prescribed format; this form means pieces of information and attributes that description has to contain of each parameter, but data format (XML, JSON),
- active part contains functions in any programming language used for communication for I4.0 components with their surroundings.

8.2. Application to Robots Mitsubishi—Full Digital Twin

The above-mentioned principles can be applied completely universally, for Mitsubishi industrial robots as well. The basis is the creation of the Administrative Shell in its active (more precisely, proactive) form. The passive part of AAS alone would certainly not be sufficient in the case of a robot-type device. From the principle of the production process, the robot is an active element that must be able to interact with other elements of production. The communication between the robot and its digital envelope itself is not limited by the AAS standard. We can use either the standardised OPC UA interface or one of the low-level approaches, such as C language libraries.

The specific implementation of the administrative shell always depends on the specific application of the given robot. In our CPF environment (Figure 1), robots are used both for assembling final products and for accurately loading semi-finished products into a CNC milling machine. In the case of an assembling robot, it is appropriate to create a digital envelope of the robot itself and implement functions for offering assembly operations of the robot in the active part of this envelope.

On the contrary, in the case of a robot that is used to load material into a machining centre, it turns out to be appropriate to create an administrative shell of the entire assembly. The robot itself here performs only a single-purpose operation, serving a single end device (milling machine). To offer and confirm this operation through a robust and administratively complex mechanism of administrative shells would be extremely inefficient. In this case, only the creation of a complete digital twin of the entire assembly (machining centre together with the robot) makes sense. Such a digital twin then offers comprehensive machining services in its active part, the parameters of which already include the time required to place the workpiece in and out of the machining centre.

On the other hand, from the point of view of monitoring the efficiency of robot movement or measuring energy consumption, it is appropriate to implement a passive administrative shell for both types of robots, which provides data on robot movement and the values of required quantities in a standardized manner.

A question for future research is whether the virtual model described in the previous chapters can be integrated into the thus created digital envelope. In terms of development complexity, this is a step that will require considerable effort. However, the result will be a truly full digital twin of the given robot, which will be able to serve both for internal control or optimisation of the robot's functions, as well as for its integration into a smart production process, control of the production flow, optimisation and response to external events.

9. Conclusions

The paper deals with mathematical modelling adapted to a specific industrial robot kinematics and dynamics, conventional control and outline of novel advanced hierarchical control concept. There are representatives of possible software tools for modelling [2,15]. They provide data (both measured and simulated) for identifying and verifying model parameters of a particular robot. Each industrial robot considered in the research requires an analysis of the manufacturer's configuration. Control design optimisation can then be designed and achieved.

The paper showed that there are many possible software tools. However, their purpose/use is different. Their use for a verification of derived models is possible, but by their combination. The use for hyper prototyping depends especially on given robot and the support in specific software tool. Key robot producers also support robots from other manufacturers, but partially. The situation is similar for the integration of robots into CPFs. Communication standards allow the user to design solutions independently of proprietary solutions.

Our future work will focus on camera vision and trajectory tracking and trajectory optimization and on further development of control based on an advanced model.

All pictured real data were captured on the assembly module (Figure 4a) involved in CPF at the College of Polytechnics Jihlava. The introduced conceptions of Industry 4.0 in Section 8 are being developed in this factory.

Author Contributions: Conceptualization, K.B. and J.J.; software, K.B. and L.V.; validation, K.B. and L.V.; formal analysis, J.J. and K.B.; investigation, K.B.; data correction, L.V. and K.B.; writing—original draft preparation, K.B. and J.J. and L.V.; writing—review and editing, K.B. and J.J. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Czech Science Foundation under the project "Controllable gripping mechanics: Modelling, control and experiments", No. 23-04676J.

Data Availability Statement: Data are contained within the article.

Acknowledgments: The authors appreciate kind support of the Czech Academy of Sciences, Institute of Information Theory and Automation and the College of Polytechnics Jihlava.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

ASS	Asset Administration Shell
CAD	Computer Aided Design
CPF	Cyber-Physical Factory
DH	Denavit–Hartenberg (concept/parameters)
DOF	Degree of Freedom
IFR	International Federation of Robotics
PID	Proportional–Integral–Derivative controller
PLC	Programmable Logic Controllers
TCP	Tool Centre Point
WR	World Robotics (statistical dept. of IFR
ZVEI	Association of the Electrical and Digital Industry

References

1. Jazar, R. *Theory of Applied Robotics*, 3rd ed.; Springer: Cham, Switzerland , 2010.
2. Raza, M.; Kumar, P.M.; Hung, D.V.; Davis, W.; Nguyen, H.; Trestian, R. A Digital Twin Framework for I4.0 Enabling Next-Gen Manufacturing. In Proceedings of the 2020 9th International Conference on Industrial Technology and Management (ICITM), Oxford, UK, 11–13 February 2020; pp. 73–77. [CrossRef]
3. Belda, K.; Jirsa, J. Control Principles of Autonomous Mobile Robots Used in Cyber-Physical Factories. In Proceedings of the 23rd International Conference on Process Control, Strbske Pleso, Slovakia, 1–4 June 2021; pp. 96–101.
4. Hendzel, Z.; Trojnacki, M. Adaptive Fuzzy Control of a Four-Wheeled Mobile Robot Subject to Wheel Slip. *Wseas Trans Syst.* **2023**, *22*, 602–612. [CrossRef]
5. Jiang, H.; Wang, Z.; Jin, Y.; Chen, X.; Li, P.; Gan, Y.; Lin, S.; Chen, X. Hierarchical control of soft manipulators towards unstructured interactions. *Int. J. Robot. Resch.* **2021**, *40*, 411–434. [CrossRef]
6. Docimo, D. A Design Framework with Embedded Hierarchical Control Architecture Optimization. In Proceedings of the 2022 American Control Conference (ACC), Atlanta, GA, USA, 8–10 June 2022; pp. 3184–3191. [CrossRef]
7. Saleem, O.; Abbas, F.; Iqbal, J. Complex Fractional-Order LQIR for Inverted-Pendulum-Type Robotic Mechanisms: Design and Experimental Validation. *Mathematics* **2023**, *11*, 913. [CrossRef]
8. Afifa, R.; Ali, S.; Pervaiz, M.; Iqbal, J. Adaptive Backstepping Integral Sliding Mode Control of a MIMO Separately Excited DC Motor. *Robotics* **2023**, *12*, 105. [CrossRef]
9. Awan, Z.S.; Ali, K.; Iqbal, J.; Mehmood, A. Adaptive backstepping based sensor and actuator fault tolerant control of a manipulator. *J. Electr. Eng. Technol.* **2019**, *14*, 2497–2504. [CrossRef]
10. Vázquez-Hurtado, C.; Altamirano-Avila, E.; Roman-Flores, A.; Vargas-Martinez, A. Towards a Mixed Virtual Reality Environment Implementation to Enable Industrial Robot Programming Competencies within a Cyber-Physical Factory. In Proceedings of the 2023 IEEE Global Engineering Education Conference (EDUCON), Kuwait, Kuwait, 1–4 May 2023; pp. 1–8. [CrossRef]
11. Gajdzik, B.; Wolniak, R. Influence of Industry 4.0 Projects on Business Operations: Literature and Empirical Pilot Studies Based on Case Studies in Poland. *J. Open Innov. Technol. Mark. Complex.* **2022**, *8*, 44. [CrossRef]
12. Grabowska, S.; Saniuk, S.; Gajdzik, B. Industry 5.0: Improving humanization and sustainability of Industry 4.0. *Scientometrics* **2022**, *127*, 3117–3144. [CrossRef] [PubMed]

13. Bilancia, P.; Schmidt, J.; Raffaeli, R.; Peruzzini, M.; Pellicciari, M. An Overview of Industrial Robots Control and Programming Approaches. *Appl. Sci.* **2023**, *13*, 2582. [CrossRef]
14. Jwo, J.S.; Lee, C.H.; Lin, C.S. Data Twin-Driven Cyber-Physical Factory for Smart Manufacturing. *Sensors* **2022**, *22*, 2821. [CrossRef] [PubMed]
15. Ryalat, M.; ElMoaqet, H.; AlFaouri, M. Design of a Smart Factory Based on Cyber-Physical Systems and Internet of Things Towards Industry 4.0. *Appl. Sci.* **2023**, *13*, 2156. [CrossRef]
16. IFR. International Federation of Robotics, Global Statistics on Industrial Robots. 2025. Available online: https://ifr.org/wr-industrial-robots/ (accessed on 1 January 2025).
17. Belda, K.; Venkrbec, L. Control Principles of Stationary Articulated Robots Used in Cyber-Physical Factories. In Proceedings of the 27th International Conference on Circuits, Systems, Communications and Computers (CSCC), Rhodes, Greece, 19–22 July 2023.
18. Belda, K.; Píša, P. Explicit Model Predictive Control of PMSM Drives. In Proceedings of the 2021 IEEE 30th International Symposium on Industrial Electronics, Kyoto, Japan, 20–23 June 2021, pp. 1–6.
19. Mitsubishi Electric. RV-FR Series, Instruction Manual, Robot Arm Setup & Maitenance, BFP-A3474-N, 2022. Available online: https://dl.mitsubishielectric.com/dl/fa/document/manual/robot/bfp-a3474/bfp-a3474r.pdf (accessed on 15 May 2024).
20. Belda, K.; Rovný, O. Predictive Control of 5 DOF Robot Arm of Autonomous Mobile Robotic System Motion Control Employing Mathematical Model of the Robot Arm Dynamics. In Proceedings of the 21st International Conference on Process Control, Strbske Pleso, Slovakia, 6–9 June 2017; pp. 339–344.
21. Belda, K. Mathematical modelling and predictive control of permanent magnet synchronous motor drives. *Trans. Electr. Eng.* **2013**, *2*, 114–120.
22. Prokop, L.; Grasblum, P. 3-Phase PM Synchronous Motor Vector Control Using a 56F80x, 56F8100, or 56F8300 Device. Freescale Semi—Conductor Application Note. 2005. Available online: https://www.nxp.jp/docs/en/application-note/AN1931.pdf (accessed on 20 January 2022).
23. Belda, K. Nonlinear Model Predictive Control Algorithms for Industrial Articulated Robots. In *Informatics in Control, Automation and Robotics: ICINCO, Revised Selected Papers*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 230–251.
24. MathWorks. *Simscape ^{TM} Multibody^{TM}, User's Guide*; MathWorks: Natick, MA, USA, 2024.
25. MathWorks. *Robotics System Toolbox^{TM}, User's Guide*; MathWorks: Natick, MA, USA, 2024.
26. Belda, K.; Dvořák, K. Path Modeling and 3D Robot Visualization for Model-Based Control of Articulated Robots. In Proceedings of the Preprints 15th European Workshop on Advanced Control and Diagnosis, Bologna, Italy, 21–22 November 2019; pp. 1–16.
27. Mitsubishi Electric. RT ToolBox3 / RT ToolBox3 mini, User's Manual, BFP-A3495-L. 2020. Available online: https://dl.mitsubishielectric.com/dl/fa/document/manual/robot/bfp-a3495/bfp-a3495u.pdf (accessed on 15 May 2023).
28. ZVEI. Reference Architecture Model Industrie 4.0 (RAMI4.0). 2015. Available online: https://www.zvei.org/fileadmin/user_upload/Presse_und_Medien/Publikationen/2016/januar/GMA_Status_Report__Reference_Archtitecture_Model_Industrie_4.0__RAMI_4.0_/GMA-Status-Report-RAMI-40-July-2015.pdf (accessed on 10 March 2023).
29. IDTA; ZVEI. Details of the Asset Administration Shell, Part 1—The Exchange of Information Between Partners in the Value Chain of Industrie 4.0 (Version 3.0RC02). 2022. Available online: https://digitalcollection.zhaw.ch/server/api/core/bitstreams/58284191-48b7-4870-8296-6ef1f3f307ad/content (accessed on 20 November 2024).
30. IDTA; ZVEI. Details of the Asset Administration Shell, Part 2—Interoperability at Runtime—Exchanging Information via Application Programming Interfaces (Version 1.0RC02). 2022. Available online: https://digitalcollection.zhaw.ch/server/api/core/bitstreams/35301a19-fade-444f-bb40-8975bad47f48/content (accessed on 10 December 2023).
31. ZVEI. Association of the Electrical and Digital Industry. 2023 Available online: https://www.zvei.org/en/ (accessed on 10 June 2024).