Available online at www.sciencedirect.com

ScienceDirect

Procedia Manufacturing 51 (2020) 258–265

Procedia MANUFACTURING

www.elsevier.com/locate/procedia

30th International Conference on Flexible Automation and Intelligent Manufacturing (FAIM2021)
15-18 June 2021, Athens, Greece.

Information model to support PLCOpen Motion Control programming from Mechanical Design

J. Garrido[a,*], D. Santos[a], J. Sáez[a], D. Silva[a], E. Riveiro[a]

[a]Automation and System Engineering Department, School of Industrial Engineering, University of Vigo, Campus Lagoas-Marcosende, Vigo 36310, Spain

---

Abstract

The design of automated industrial machinery involves different areas of engineering, each of which employs its own and different information representation systems and software tools. The lack of a common information model to collect and organize the essential common data of each technology, prevents collaborative multidisciplinary engineering work, which complicates the use of a mechatronic approach. This article proposes the structure of an information model that allows to include geometric, kinematic and logical information related to the tools and objects that the machine manipulates, organized hierarchically according to the mechanical structure of the machine. This model complements an earlier development by the authors by calling MMCS "Mechanical and Motion Control Schematics", which focuses on graphical representation. By combining them, the dynamic behavior can be visualized together.

Keywords: Mechatronic ; PLCOpen ; Motion Control ; Standards ; Information Model

---

1. Introduction

Machines design process combines knowledge from different areas. In the case of machines with servo axes, the interaction and interdependence between mechanics and control can be very high. Moreover, with the arrival of new machine controllers, new electronic devices and new automation development software, new mechatronics concepts have appeared in the field of machine design. As defined by Mechatronics Elsevier journal editorial board [1], "Mechatronics is the synergistic combination of precision mechanical engineering, electronic control and systems thinking in the design of products and manufacturing processes". This complexity increases with the possibility of defining temporal and electronic kinematic relations between axes, as for instance master-slave relationships, CAM table dependencies, etc., as well as the use of virtual axes associated with real axes for control reasons.

The first design decisions in each technological point of view (mechanical and control) affects the other. As the design progresses, specific tools for each technology and specific standards come into action. Design systems are too much oriented to their specific technological point of view. These systems manage information of different natures, which makes it difficult to combine it into a single representation, even though both fields are complementary. On the one hand, different types of technical representations, such as drawings, can show the mechanical information of a machine, while on the other hand, the movements of servomechanisms or servo axes that drive the machine are described by text commands or "time-based" representations that will become the motion controller program.

Starting from the functional requirements, in each step more detailed information is progressively added. Design tools also become specific. Both technological points of view, mechanical and control, hardly include relevant information from another field [2]. This disjunction has been a classical communication problem between mechanical designers and automation software designers, which should be traced up to the mechanic oriented or electronic oriented education programs. A similar communication barrier has been traditionally present at machine design education environments, where graphics and machine parts representations from machines mechanical

---

* Corresponding author. Tel.: +34-986-812-610.
E-mail address: jgarri@uvigo.es (J. Garrido).

10.1016/j.promfg.2020.10.037

perspective have been very different from graphics for an electronic and control perspective. There is no standard for the joint representation of machines with servo axes and its corresponding motion control, and informal mechatronic representations are used in scientific, industry and therefore educational fields, as it will be reviewed in next section 2. A graphic representation system combining mechanical and motion control information may be useful to cover the gap between current standards. To achieve it, MMCS stands for "Mechanical and Motion Control Schematics" is being developed by the authors, and a more detailed explanation may be found in [2]. MMCS will be briefly introduced in section 3. The fourth section continues with the description of the MMCS digital information model to support MMCS digital design instantiation, and the MMCS extension to represent machine movements sequence as a way to specify the motion control program. Finally, section 5 is an example using that digital movement's specification. The article ends with conclusions and future lines.

2. Mechanic and Electronic design representation methods and standards

2.1. Mechanical point of view: Technical drawing

Different graphic representation systems are used in architecture, topography, electrics and many other branches of engineering, including mechanics. For example, the standard ISO128 [2] Part 20 establishes basic conventions for lines, ISO 5455 [3] deals with scales, and ISO 6433 [4] with part references. Drawings are usually classified into sketches or blueprints. Projections or views of the objects are used with scales. Through the use of measures and other symbols, further information is added.

2.1.1. Mechanical drawings

There are different types of mechanical drawings. For example, in mechanical engineering, a blueprint for the manufacture of a component will also include manufacturing tolerance, finishes, materials, etc. The specific details of these components are fundamental for mechanical design, but not for their control. Furthermore, this type of drawings is a representation with a too static approach, as mobile elements are not clearly identified. The same happens with the reference systems and the measures used to describe the movement. For example, the coupling of a mobile part with those it moves has to be deduced from the interpretation of the manufacturing and assembly of said part. This entails a certain difficulty if one is not familiar with these types of drawings or if the machine is a complex one.

This type of drawings is restricted to geometric conditions and it does not establish kinematic or dynamic conditions. Electric components such as limit switches or homing sensors can be represented in their mechanical version, but without specifying their function or identification according to electric standards.

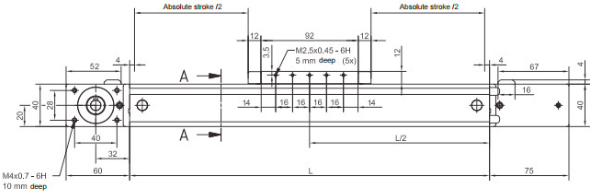
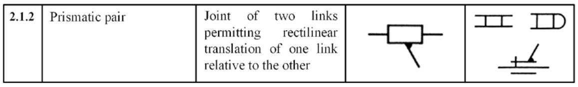


Fig. 1. Side view of a mechanical drawing for a belt driven linear module.



2.1.2	Prismatic pair	Joint of two links permitting rectilinear translation of one link relative to the other		

Fig. 2. Examples of kinematic diagrams according to the standard ISO 3952.

Figure 1 shows the mechanical drawing of a commercial belt driven linear module [5]. The information relevant to the control system is limited to:

- The absolute stroke, in this case, measure L minus the width of the mobile carriage.
- The specific point of the mobile carriage or tool that will serve as a reference to measure.

2.1.2. Kinematic schemes

The standard ISO 3952 [6] is a system of graphic symbols for the simplified representation of rigid solids and the mechanical relationships between them which define a mechanism, without taking into account construction details such as those that can be found in mechanical drawings. Typical examples of this standard can be seen in Figure 2 which shows the symbols proposed in this standard for a linear module like the one in Figure 1 above.

This simplification eliminates important information for detailed design and manufacturing, such as dimensions, measures, finishes, etc. The reference and coordinate systems in this standard are used for mathematic modelling purposes. Examples of this symbol representation combined in applications of servo drives can be found in the literature [7,8]. It is also used for the schematic representation of CNC, such as for example in [9].

Despite being aimed at kinematics and system dynamics, it lacks the symbols to include temporal links between mobile parts created by the control system. It adds unnecessary details for the control system, such as the kinds of kinematics pairs that exist between the linkages of mechanical elements. It is not as widely used as in the case of mechanical drawings. Articles can be found where a schematic representation of linear axes is used with the addition of necessary details for the presentation of the ideas to discuss. [10–14].

2.2. Programming languages for multi-axis control, PLCOpen for motion control.

There are standardized programming languages to implement the control sequences of a machine and the motion commands to its servo axes, which may be divided in the standards for machine tools programming (CNC), and the standards for automated industrial processes controlled by PLC's.

ISO6983 [11] is used to describe the trajectories of the tool in CNC machines. The sequence of movements is described by a succession of text codes with the prefix ”G” that includes coordinates, velocity parameters and other details. Its interpretation is not evident since it is oriented to its direct processing by motion controllers. New standards, such as STEP-NC [10], consider more detailed manufacturing information of the piece. This type of standards is especially tough for CNC machines with specific axes configurations. However, they are not suitable for another type of automated machines, although they also perform motion sequences, with a wide range of possible applications and automated by PLC (Programming Logic Controllers). The sequences of movements are described by instructions defined by programming languages. There are numerous solutions to implement and describe the operation of machines using those languages. The most important standard is IEC 61131-3 [14], which suggests different programming languages: Ladder Diagram (LD), Structured Text (ST), Instruction List (IL), Function Block Diagram (FBD) and Sequential Flow Chart (SFC). Its use is widespread and widely accepted, not only by users but also by equipment manufacturers.

IEC 61131-3 was adopted by PLCOpen [15], who expanded it by specifying, among other things, a set of libraries of function blocks for axis control [16]. It defines a set of FBs (function blocks) to program the control of servo drives, although it does not implement it. It comprises from simple movements PTP (point to point), to complex coordinated movements which create virtual/logical relationships between axes, equivalent to their mechanic counterparts, such as mechanical cams and others. These software relationships may be activated and altered during the operation of the machine, changing the logical state of the axes with effect on the mechanics. There is also a similar state machine for the case of a group of axes.

The behaviour of the machine's axes is the result of the execution of a sequence of these instructions by a PLC. The program comments could include information about what it does, but if the mechanics were complex, the description would be complicated and prone to wrong interpretations without some kind of mechanical drawing or an equivalent.

The interpretation of the source code of a program with these characteristics can be very complicated, even for specialists in the field. It is not linked to any kind of graphic representation, except graphic records of key axis parameters, such as position, speed and torque. Furthermore, the instructions include irrelevant information to the mechanical system, such as the names of logical instances, variables, data types, details of the instruction execution, etc.

Commands can also be executed against virtual axes, i.e. logical axes that are not linked to an actual physical servo drive.

2.3. *Other representations: Scientific Literature and Technical Documentation and Manuals*

Informal representations are very commonly used in scientific literature to represent machines with servo axes and motion control. The following are some examples where axes and names have been drawn on photos of the machine, as in [17–19]. Examples of computer-generated images of the machine are also found, as in [20–24]. Simplified drawings are found, as in [25–29]. Even the PLCOpen standard itself makes use of this type of drawings [30]. Also, examples of trajectories representations can be found in [31] and [21].

2.3.1. *Technical Documentation and Manuals*

The manufacturers of components for machinery regularly make use of technical drawing standards in the technical documentation and manuals of their products. However, non-standard or informal drawings and schemes can be found in the documentation of motion controllers or servomotors, for example in [32–34] together with explanations and examples of the use of motion commands.

3. Mechatronic Motion Control Schemes Model

There is no standard for the joint representation of machines with servo axes and its corresponding motion control, and informal mechatronic representations are used in scientific, technological and therefore educational fields. A new graphic representation system combining mechanical and motion control information may be useful to cover the gap between current standards (Figure 3). The authors of this research have already proposed the basis for a new type of schemes for the joint representation of mechatronic and motion control systems or MMCS (Mechatronic Motion Control Schematics) [35].

MMCS schematics is the common zone between mechanical design and programming (Figure 3). As a representative example, Figure 4 is the MMCS top view of a ball screw driven linear module of Figure 1. This linear axis proposed simplification takes advantages of some graphic elements of ISO 3952 [6], such as parallel lines for fixed points and vertical lines to delimit the path. The proportions should be maintained to facilitate identification of the simplified symbol with the actual element. Although from the point of view of control, a point is positioned, the representation of the mobile carriage is maintained. An important characteristic of MMCS axis representation is that it is clearly stated the mobile part (as a white rectangle), from the guide element (a black rectangle). In Figure 4, left (a) configuration is completely different, in terms of movement than the right one (b). In (a), white box is the mobile part and the black axis is fixed, while on the right, the black box is fixed and the white axis is the mobile part.

MMCS schematics is the common zone between mechanical design and programming (Figure 3). As a representative example, Figure 4 is the MMCS top view of a ball screw driven linear module of Figure 1. This linear axis proposed simplification takes advantages of some graphic elements of

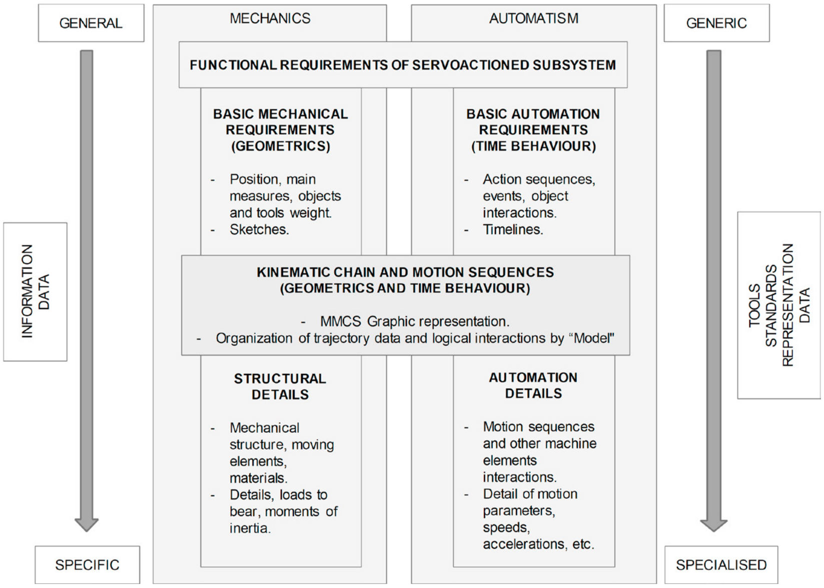


Fig. 3. MMCS model information map.

ISO 3952 [6], such as parallel lines for fixed points and vertical lines to delimit the path. The proportions should be maintained to facilitate identification of the simplified symbol with the actual element. Although from the point of view of control, a point is positioned, the representation of the mobile carriage is maintained. An important characteristic of MMCS axis representation is that it is clearly stated the mobile part (as a white rectangle), from the guide element (a black rectangle). In Figure 4, left (a) configuration is completely different, in terms of movement than the right one (b). In (a), white box is the mobile part and the black axis is fixed, while on the right, the black box is fixed and the white axis is the mobile part.

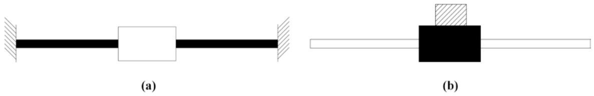


Fig. 4. Schematic view of linear actuator. a) Larger guideway than moving carriage, b) Smaller guideway than moving carriage.

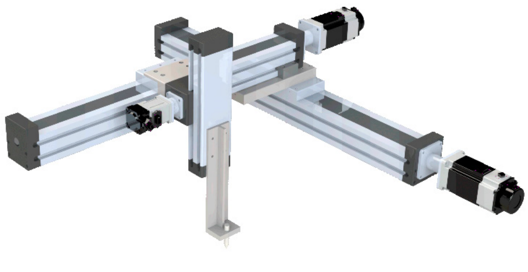


Fig. 5. Cartesian system 3D view.

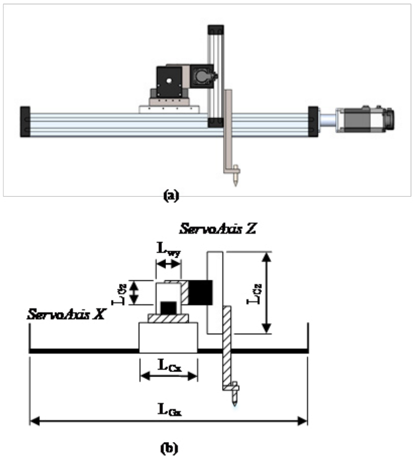


Fig. 6. Side view of the Cartesian system. (a) 3D Version. (b) Equivalent version in MMCS.

Figure 5 shows the Cartesian system used as an example to show how this type of system can be represented with the MMCS proposal (Figure 6).

The connecting pieces between the servo axes are represented with a simplified shape and filled with a striped pattern. The tool is also represented in the same way. The active or mobile elements are represented in white, while the thick black lines represent the guides and stops. The servo axes X and Y are similar to the one studied in the previous section and, therefore, their simplifications are reused to represent them. Servo axis Z has a mechanically different configuration if compared to X and Y. It can be seen in Figure 6 that the mobile element or carriage has a larger size than the fixed element.

Figure 6 shows the side view of the 3D model of Figure 5. In the case of servo axis Z, it can be seen in Figure 8 that the vertical white rectangle representing the mobile element is larger than the black rectangle representing the guide. Simplified versions of the joining pieces and the tool can be identified by their filling with striped pattern.

4. Motion and logic information model for Mechatronic Motion Control Schemes extension

With the MMCS approach, a joint design stage for mechanics and programmers, with a common language, would be possible. The resulting design is the starting point for the next steps of making detailed designs for each technological point of view.

However, the MMCS designs represent a static vision of the moving elements of the machine and the specification of its relationship with the guide elements and coupling ones. Nevertheless, the temporal representation of these movements is still missing, both in terms of trajectories and in the interaction with the other non-motion elements of the machine. For this, an information model of representation of the trajectories (toolpaths) that result from the movements of the axes is proposed. Also, the new types of axes (virtual axes, encoder and generated axes) are taken into consideration, as well as temporary couplings present in the new motion programming standards such as electronic cams, fly cutting, position/speed synchronization, etc.

Figure 7 shows the information model generic structure. Main model classes are described below: MACHINE_MOTION_CONTROL_LOGIC: Model main class. It relates the motion control information and the associated machine logic. This class groups the following three subclasses:

- MMCS_DATA: It contains the MMCS schematic specification of the machine. A complex machine might need different MMCS for different parts with different configurations: Cartesian, Delta, etc.
- SYSTEM_COORDINATES: It contains the data to associate the mechanics to the different coordinate systems. Two subclasses can be distinguished: one for the machine TCP's (Tool Center Point), and other for each one of the OCP's (Object Center Point); processed objects by the machine and other auxiliary objects as, for instance, sensors.
- MOTION_SEQUENCES_TCP: This class collects information about dynamic or control logic. Motion sequences are the combination of individual movements and tools actions, as well as their initial conditions and associated logic. For example, in the case of a palletizing machine, instances of this class could be: ”Pick a box at the input position”, ”Place the box at the pallet”, ”Move to maintenance Position”, etc. This class is made up of three other classes, AXIS_INITIAL_CONDITIONS, LOGIC_INITIAL_CONDITIONS and MOVEMENT. AXIS_INITIAL_CONDITIONS describes the initial conditions that must be met by the axes. LOGIC_INITIAL_CONDITIONS specifies the logical state of the machine to start a sequence of movements. For example, the sequence “Pick a Box at the input position” requires servo axes are activated, the tool is in the waiting area next to the entrance of the boxes, a box is in the loading area, etc. MOVEMENT describes the individual movements of a sequence. In turn, each movement has its INITIAL_CONDITIONS and AXIS_MOVEMENT. For example, ”Pick a Box at the input” can be broken down into the following movements: ”Fast movement up to approaching position”, ”Slow approach up to the box top”, ”Slow box raising to safety height”, ”Box movement to pallet zone”, etc.

The model representation technology is depending on the used for persistent models instantiation. Initially, the research has opted for XML schema technology for the representation of designs in XML files (8). A detail of the schema can be seen in Figure 8, while Figure 9 is the corresponding XML file detail.

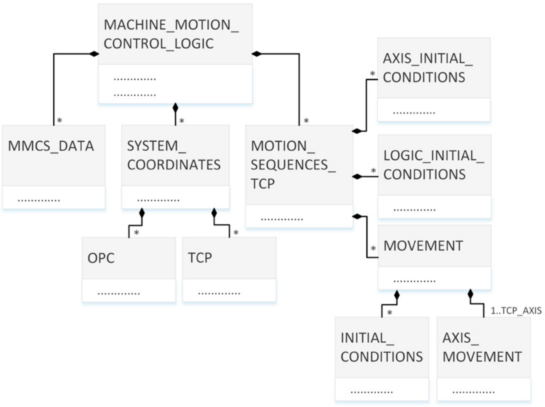


Fig. 7. MMCS information model general structure.

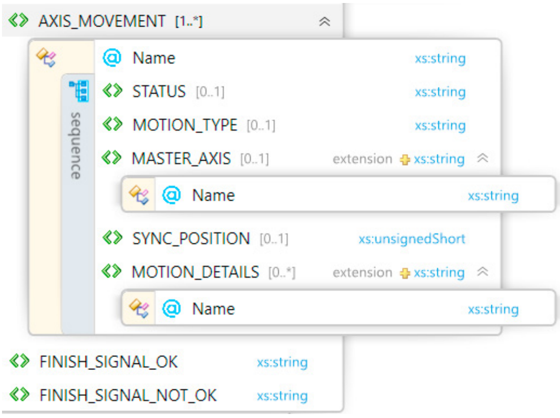
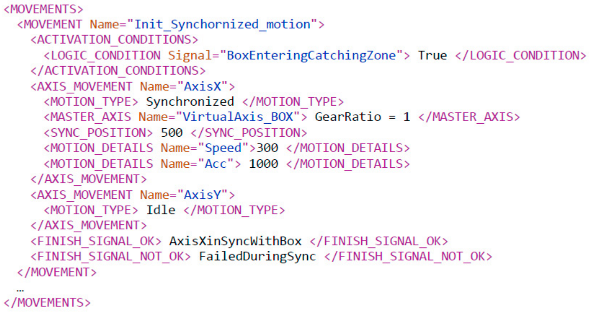


Fig. 8. MMCS Schema model detail.



```xml
<MOVEMENTS>
  <MOVEMENT Name="Init_Synchornized_motion">
    <ACTIVATION_CONDITIONS>
      <LOGIC_CONDITION Signal="BoxEnteringCatchingZone"> True </LOGIC_CONDITION>
    </ACTIVATION_CONDITIONS>
    <AXIS_MOVEMENT Name="AxisX">
      <MOTION_TYPE> Synchronized </MOTION_TYPE>
      <MASTER_AXIS Name="VirtualAxis_BOX"> GearRatio = 1 </MASTER_AXIS>
      <SYNC_POSITION> 500 </SYNC_POSITION>
      <MOTION_DETAILS Name="Speed">300 </MOTION_DETAILS>
      <MOTION_DETAILS Name="Acc"> 1000 </MOTION_DETAILS>
    </AXIS_MOVEMENT>
    <AXIS_MOVEMENT Name="AxisY">
      <MOTION_TYPE> Idle </MOTION_TYPE>
    </AXIS_MOVEMENT>
    <FINISH_SIGNAL_OK> AxisXinSyncWithBox </FINISH_SIGNAL_OK>
    <FINISH_SIGNAL_NOT_OK> FailedDuringSync </FINISH_SIGNAL_NOT_OK>
  </MOVEMENT>
  …
</MOVEMENTS>
```

Fig. 9. MMCS XML design detail.

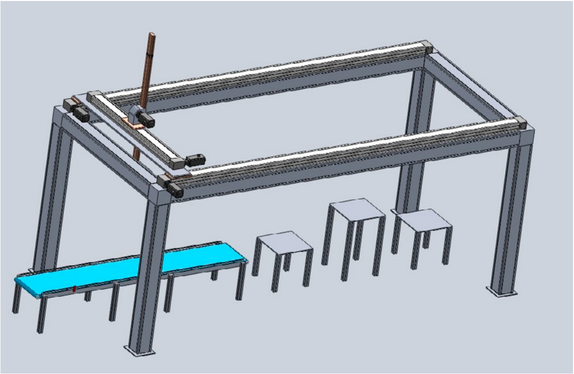


Fig. 10. 3D view of the machine example application.

5. Example

This section presents an example of the application of the MMCS and model information to the motion control sequences design of a specific machine. Figure 10 shows the 3D machine model.

The machine consists of a set of linear axes in the Cartesian configuration. Main horizontal axis “X”, carries a vertical one “Z” with a tool capable of holding and releasing a box. The ”Y” axis that laterally moves ”Z” axis is not used in this example. The purpose of the machine is to pick boxes coming from an input belt and place them in an exit table, where another machine removes them to continue with the process. To perform “on the fly” picking requires that the X-axis has to be synchronized in speed and position with the input belt while picking the box. And previously, the TCP has to be aligned on the centre of the box before the axis descends to pick it with the clamping tool. Figure 11 depicts the MMCS graphic schematic representation of the process.

The home position of the TCP corresponds to the travel limits of the X and Z axes. Three main movement sequences can be identified: ”Catching_Box”, ”Box_To_Table”, ”Returning_to_waiting_position”. The instantiated data of the sequences, according to the model would be as seen in Figure 12.

Taking “Catching_Box” as an example, its initial conditions are shown in Figure 12, under the “<INITIAL_CONDITIONS>” entity. The sequence ”Catching_Box” could be segmented according to the movements represented in Figure 11. Their movements are identified with the symbols #1, #2, #3, #4, #5 on the path of the

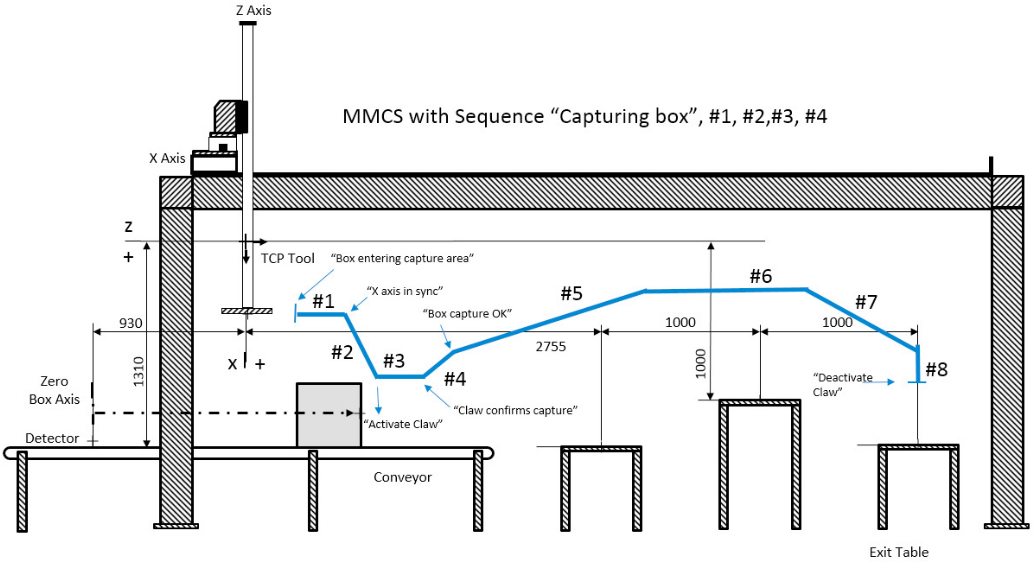


Fig. 11. MMCS representation of the machine example with the main tool trajectory for “Catching_Box” sequence. Dash line represents box trajectory as a virtual axis.

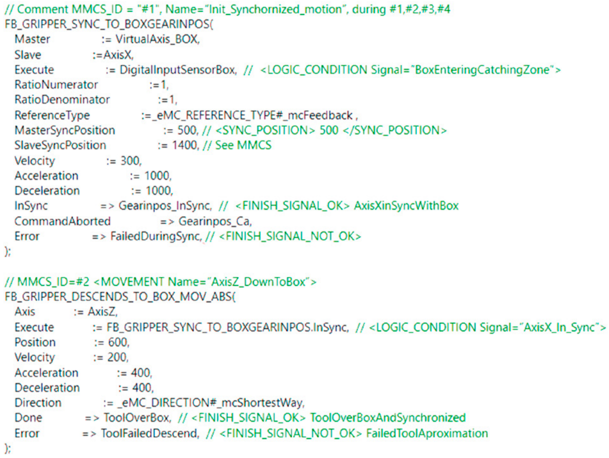
TCP (continuous line). Data for movements details can be seen in Figure 13.

```xml
<MACHINE_MOTION_LOGIC>
  <MMCS> … </MMCS>
  <TCPS> … </TCPS>
  <COORDINATE_SYSTEM> … </COORDINATE_SYSTEM>
  <MOTION_LOGIC_SEQUENCES>
    <MOTION_LOGIC_SEQUENCE Id="0001" Name="Catching_Box">
      <EXECUTION_TIME_REQUIRED> 4 </EXECUTION_TIME_REQUIRED>
      <COORD_SYSTEM_USED> MachineMainCoordinateSystem </COORD_SYSTEM_USED>
      <INITIAL_CONDITIONS>
        <AXIS_CONDITION Axis="AxisX">
          <POWER> ON </POWER>
          <STATUS> IDLE </STATUS>
          <MAX_POSITION> 110 </MAX_POSITION>
          <MIN_POSITION> 100 </MIN_POSITION>
        </AXIS_CONDITION>
        <AXIS_CONDITION Axis="AxisY"> … </AXIS_CONDITION>
        <LOGIC_CONDITIONS>
          <LOGIC_CONDITION Name="GripperOpen"> True </LOGIC_CONDITION>
          <LOGIC_CONDITION Name="GripperOK"> True </LOGIC_CONDITION>
          <LOGIC_CONDITION Name="AutomaticMode"> True </LOGIC_CONDITION>
          …
        </LOGIC_CONDITIONS>
      </INITIAL_CONDITIONS>
      <MOVEMENTS> … </MOVEMENTS>
    </MOTION_LOGIC_SEQUENCE>
    …
    <MOTION_LOGIC_SEQUENCE Id="0002" Name="Box_to_table"> … </MOTION_LOGIC_SEQUENCE>
    <MOTION_LOGIC_SEQUENCE Id="0003" Name="Returing_to_waiting_position"> …
    </MOTION_LOGIC_SEQUENCE>
  </MOTION_LOGIC_SEQUENCES>
  …
</MACHINE_MOTION_LOGIC>
```

Fig. 12. XML example, from <MACHINE_MOTION_LOGIC> to <MOTION_LOGIC_SEQUENCE>.

```xml
<MOVEMENTS>
  <MOVEMENT Name="Init_Synchornized_motion">
    <MMCS ID="1"> </MMCS>
    <ACTIVATION_CONDITIONS>
      <LOGIC_CONDITION Signal="BoxEnteringCatchingZone"> True </LOGIC_CONDITION>
    </ACTIVATION_CONDITIONS>
    <AXIS_MOVEMENT Name="AxisX">
      <MOTION_TYPE> Synchronized </MOTION_TYPE>
      <MASTER_AXIS Name="VirtualAxis_BOX"> GearRatio = 1 </MASTER_AXIS>
      <SYNC_POSITION> 500 </SYNC_POSITION>
      <MOTION_DETAILS Name="Speed"> 300 </MOTION_DETAILS>
      <MOTION_DETAILS Name="Acc"> 1000 </MOTION_DETAILS>
    </AXIS_MOVEMENT>
    <AXIS_MOVEMENT Name="AxisY">
      <MOTION_TYPE> Idle </MOTION_TYPE>
    </AXIS_MOVEMENT>
    <FINISH_SIGNAL_OK> AxisXinSyncWithBox </FINISH_SIGNAL_OK>
    <FINISH_SIGNAL_NOT_OK> FailedDuringSync </FINISH_SIGNAL_NOT_OK>
  </MOVEMENT>
  <MOVEMENT Name="AxisZ_DownToBox">
    <MMCS ID="2"> </MMCS>
    <ACTIVATION_CONDITIONS>
      <LOGIC_CONDITION Signal="AxisX_In_Sync"> True </LOGIC_CONDITION>
    </ACTIVATION_CONDITIONS>
    <AXIS_MOVEMENT Name="AxisX">
      <STATUS> No changes </STATUS>
    </AXIS_MOVEMENT>
    <AXIS_MOVEMENT Name="AxisZ">
      <MOTION_TYPE> Discrete </MOTION_TYPE>
      <MOTION_DETAILS Name="Position"> 600 </MOTION_DETAILS>
      <MOTION_DETAILS Name="Speed"> 200 </MOTION_DETAILS>
      <MOTION_DETAILS Name="Acc"> 400 </MOTION_DETAILS>
      <MOTION_DETAILS Name="SpeedProfile"> Trapezoidal </MOTION_DETAILS>
    </AXIS_MOVEMENT>
    <FINISH_SIGNAL_OK> ToolOverBoxAndSynchronized </FINISH_SIGNAL_OK>
    <FINISH_SIGNAL_NOT_OK> FailedToolAproximation </FINISH_SIGNAL_NOT_OK>
  </MOVEMENT>
  <MOVEMENT Name="Gripper_activation">
    <MMCS ID="3"> </MMCS>
    ...
  </MOVEMENT>
  <MOVEMENT Name="Z_Axis_Elevate_box">
    <MMCS ID="4"> </MMCS>
    ….
  </MOVEMENT>
  <MOVEMENT Name="Finishing_Catching_phase">
    <MMCS ID="5"> </MMCS>
    ....
  </MOVEMENT>
</MOVEMENTS>
```

Fig. 13. XML, deeper detail, with first two movements of the “Catching_Box” sequence.



```
// Comment MMCS_ID = "#1", Name="Init_Synchornized_motion", during #1,#2,#3,#4
FB_GRIPPER_SYNC_TO_BOXGEARINPOS(
  Master              := VirtualAxis_BOX,
  Slave              :=AxisX,
  Execute              := DigitalInputSensorBox,  //  <LOGIC_CONDITION Signal="BoxEnteringCatchingZone">
  RatioNumerator              :=1,
  RatioDenominator              :=1,
  ReferenceType              :=_eMC_REFERENCE_TYPE#_mcFeedback ,
  MasterSyncPosition              := 500, // <SYNC_POSITION> 500 </SYNC_POSITION>
  SlaveSyncPosition              := 1400, // See MMCS
  Velocity              := 300,
  Acceleration              := 1000,
  Deceleration              := 1000,
  InSync              => Gearinpos_InSync,  //  <FINISH_SIGNAL_OK> AxisXinSyncWithBox
  CommandAborted              => Gearinpos_Ca,
  Error              => FailedDuringSync, // <FINISH_SIGNAL_NOT_OK>
);

// MMCS_ID=#2 <MOVEMENT Name="AxisZ_DownToBox">
FB_GRIPPER_DESCENDS_TO_BOX_MOV_ABS(
  Axis          := AxisZ,
  Execute          := FB_GRIPPER_SYNC_TO_BOXGEARINPOS.InSync,  // <LOGIC_CONDITION Signal="AxisX_In_Sync">
  Position          := 600,
  Velocity          := 200,
  Acceleration          := 400,
  Deceleration          := 400,
  Direction          := _eMC_DIRECTION#_mcShortestWay,
  Done          => ToolOverBox,  // <FINISH_SIGNAL_OK> ToolOverBoxAndSynchronized
  Error          => ToolFailedDescend,  // <FINISH_SIGNAL_NOT_OK> FailedToolAproximation
);
```

Fig. 14. Structured Text implementation for movements “Init_Synchronized_motion” and “AxisZ_DownToBox” of sequence “Catching_Box”.

This movement data can be used as a starting point to implement the generation of source code for motion control programs. Figure 14 is the code written using the programming language “Structured Text” (PLCOpen standard) for the ”Init_Synchornized_motion” and ”AxisZ_DownToBox” movements.

There does not have to be a direct correspondence between each <MOVEMENT> element with an instruction or movement order. For example, PLCOpen motion function block to implement synchronization of a slave axis concerning a master axis includes both the synchronism search phase and the synchronized movement phase. However, in this example, movement #1 corresponds to the synchronization search phase and #2, #3, #4 with those of synchronized movement.

Concerning the Z-axis, during #1 and #3, it does not execute any movement and in #2 and #3, it executes two movements of discrete type or PointToPoint. Finally, movement #5 could correspond to a coordinated movement of the X and Z axes.

6. Conclusions and Future work

MMCS model and its motion and logic provide a way for an integrated representation of mechanical, motion control and associated logic machine design information. The article has presented an information model that enables the graphical representation of mechanics, including trajectories, to be linked with logical information, which can be the common starting point for the design and implementation of the mechanical system and automation.

With this information model, a continuous digital path from the first phases of the machine’s conceptual design to the effective implementation of the source code of the motion controller is available It represents an intermediate point between the operation and the code implementation. It also provides the basis for the integrated visualization by a

3D mechanical design environment of trajectories and axes movements that generate them. In addition, the model allows linking manufacturing resources with motion elements.

As future lines, other motion control functionalities not covered yet with this model will be explored, such as parameterizable routes, parameterizable movements based on run time sensor readings, etc. Incorporate more motion functionalities such as cams, torque control, and in general, the advanced functionalities defined by the different parts of PLCOpen Motion Control, for instance, Function Blocks for tracking parts detected by a vision system while moving on a conveyor belt.

References

[1] Elsevier, Mechatronics, The Science of Intelligent Machines, (n.d.). https://www.journals.elsevier.com/mechatronics.

[2] ISO, 128:2003 Technical drawings. . General principles of presentation, parts 1-2, (2003).

[3] ISO, 5455:1979 Technical drawings. Scales, (1979).

[4] ISO, 6433:2012 Technical product documentation. Part references, (2012).

[5] UNIMOTION, “MTJ 65 Linear Unit”, (n.d.). http://unimotion.eu/en/binary/sapgetfile/nr/00000000000000000001071776

[6] ISO, 3952:1981 Kinematic Diagrams. Graphical symbols, parts 1-4, (1981).

[7] H. Li, Y. Zhang, Seven-bar mechanical press with hybrid-driven mechanism for deep drawing; Part 1: Kinematics analysis and optimum design, J. Mech. Sci. Technol. 24 (2010) 2153–2160.

[8] W. Tian, W. Gao, D. Zhang, T. Huang, A general approach for error modeling of machine tools, Int. J. Mach. Tools Manuf. 79 (2014) 17–23.

[9] ISO, 230-2:2016 Test code for machine tools. Part 2: Determination of accuracy and repeatability of positioning of numerically controlled axes, (2016).

[10] ISO, 14649-1:2003 Industrial automation systems and integration – Physical device control – Data model for computerized numerical controllers – Part 1: Overview and fundamental principles, (2003).

[11] ISO, 6983-1:2009 Automation systems and integration – Numerical control of machines – Program format and definitions of address words – Part 1: Data format for positioning, line motion and contouring control systems, (2009).

[12] ISO, 841:2001 Industrial automation systems and integration. Numerical control of machines. Coordinate system and motion nomenclature, (2001).

[13] ISO, 369:2009 Machine tools. Symbols for indications appearing on machine tools, (2009).

[14] IEC, 61131-3:2013 Programmable controllers - Part 3: Programming languages, (2013).

[15] PLCOpen, PLCOpen, (2019). https://www.plcopen.org.

[16] PLCOpen, Function Blocks for Motion Control, 2.0, (n.d.).

[17] K.S. Sollmann, M.K. Jouaneh, D. Lavender, Dynamic modeling of a two-axis, parallel, H-frame-type XY positioning system, IEEE/ASME Trans. Mechatronics. 15 (2010) 280–290.

[18] B.D. Bui, N. Uchiyama, K.R. Simba, Contouring control for three-axis machine tools based on nonlinear friction compensation for lead screws, Int. J. Mach. Tools Manuf. 108 (2016) 95–105.

[19] T. Tang, H. Fang, J. Zhang, Hierarchical design, laboratory prototype fabrication and machining tests of a novel 5-axis hybrid serial-parallel kinematic machine tool, Robot. Comput. Integr. Manuf. 64 (2020) 101944.

[20] D.G. Lee, S. Oh, H. Il Son, Maintenance Robot for 5-MW Offshore Wind Turbines and its Control, IEEE/ASME Trans. Mechatronics. 21 (2016) 2272–2283.

[21] X. Wang, H. Gao, O. Kaynak, W. Sun, Online Deflection Estimation of X-Axis Beam on Positioning Machine, IEEE/ASME Trans. Mechatronics. 21 (2016) 339–350.

[22] J. Yang, Y. Altintas, Generalized kinematics of five-axis serial machines with non-singular tool path generation, Int. J. Mach. Tools Manuf. 75 (2013) 119–132.

[23] N. Borchardt, R. Hinzelmann, D.S. Pucula, W. Heinemann, R. Kasper, Winding machine for automated production of an innovative air-gap winding for lightweight electric machines, IEEE/ASME Trans. Mechatronics. 21 (2016)

[24] J. Zhang, Y. Zhao, Y. Jin, Kinetostatic-model-based stiffness analysis of Exechon PKM, Robot. Comput. Integr. Manuf. 37 (2016) 208–220.

[25] J. Zhao, Y. Liu, Q. Li, Research on error synthetical modeling of multi-functional composite five-axis CNC machine tool, in: 2016 IEEE Int. Conf. Mechatronics Autom. IEEE ICMA 2016, 2016: pp. 1814–1821.

[26] L. Guo, M. Tomizuka, High-speed and high-precision motion control with an optimal hybrid feedforward controller, IEEE/ASME Trans. Mechatronics. 2 (1997) 110–122.

[27] Y. Qin, J. Xiao, G. Wang, The open architecture cnc system based on 6-axis flame pipe cutting machine, in: Proc. - 3rd Int. Conf. Meas. Technol. Mechatronics Autom. ICMTMA 2011, 2011: pp. 878–881.

[28] K.Y. Chen, Sliding mode minimum-energy control for a mechatronic motor-table system, IEEE/ASME Trans. Mechatronics. 21 (2016) 1487–1495.

[29] W. Liping, X. Huayang, G. Liwen, Z. Yu, A novel 3-PUU parallel mechanism and its kinematic issues, Robot. Comput. Integr. Manuf. 42 (2016) 86–102.

[30] PLCOpen, Application Examples with PLCOpen Motion Control, (n.d.).

[31] M. Moghaddam, S.Y. Nof, Parallelism of Pick-and-Place operations by multi-gripper robotic arms, Robot. Comput. Integr. Manuf. (2016).

[32] Omron Corporation, NJ/NX-series Motion Control Instructions Reference Manual (W508), (2017) 3–282.

[33] B.A.G.& C. KG, TwinCAT 3 PLC Lib: Tc2_MC2 Manual, (2016) 74.

[34] 3S. Smart Software Solutions GmbH, Softmotion in CoDeSys 2.3 User Manual, (2009).

[35] J.Garrido, D.S. Esterán, New motion control machine elements representation for mechatronic education, Procedia Manuf. 38 (2019) 686–693.