SOLIDWORKS ADD-IN FOR MOTION SIMULATION, LAYOUT ANALYSIS AND COLLISION DETECTION OF SUBSTRATE HANDLING ROBOTS

Nikolay Bratovanov

Abstract: *The paper is dedicated to the implementation of a tool for motion simulation, layout analysis and collision detection of substrate handling robots in the form of add-in for SolidWorks. The main goal is developing a system that allows the direct usage of standard SolidWorks robot 3D models for simulation, offline programming, analysis and optimization of automated cells, applicable in the field of semiconductor device manufacturing. The resultant system allows the simultaneous execution of design, simulation and evaluation activities, based on a single software platform, thus increasing productivity, saving time and reducing costs. The proposed add-in concept contributes to achieving a much more integrated appearance and behavior of the tool, enhancing its functionality, and eliminating issues associated with reliability and security.*

Keywords: *motion simulation; layout analysis; collision detection; substrate handling robots; kinematic modeling; robot offline programming; SolidWorks add-in;*

1. Introduction

Regarded as one of the most challenging and complex production systems that involve huge capital investment and advanced automation technologies, semiconductor device manufacturing spans across many different stages, including manufacturing of silicon wafers, where electronic circuitry is layered, chip manufacturing that involves circuit probing, and product manufacturing from which the end integrated circuit-based semiconductor products are finally assembled and tested. Since fabrication of semiconductor devices demands sophisticated control on quality, variability, yield, and reliability, it is crucial that all manufacturing processes are highly automated [1].

The existing manufacturing technologies for semiconductor integrated circuits and flat panel displays include processing of silicon wafers and glass panels, referred to as substrates, via automated atmospheric/vacuum cluster tools, which are serviced by one or more substrate handling robots. Typical operations performed by these robots include elementary rotational and straight-line moves, which are often combined and blended into more complicated planar or three-dimensional trajectories, in order to comply with complex workspace geometries [2]. Substrate handling robots are essential elements of semiconductor industry automation as they are intensively utilized for transferring wafers between the different tool processing stations within the front-end and back-end minienvironments. Such an automated approach plays an important role in the overall semiconductor manufacturing as wafers have to be transferred accurately, rapidly and smoothly between different process stations, while at the same time strict requirements for cleanliness and particle generation have to be observed. It is therefore reasonable to consider substrate handling robots as key tool automation components, whose performance, accuracy and reliability are essential for the whole semiconductor manufacturing process. Hence, the ability to study, simulate, analyze and optimize the mobility and behavior of substrate handling robots via powerful tools for motion simulation, offline programming, layout analysis and other beneficial activities would inevitably result in increased throughput, improved final product quality and enhanced overall performance of the semiconductor fabs (Fig. 1) [3].

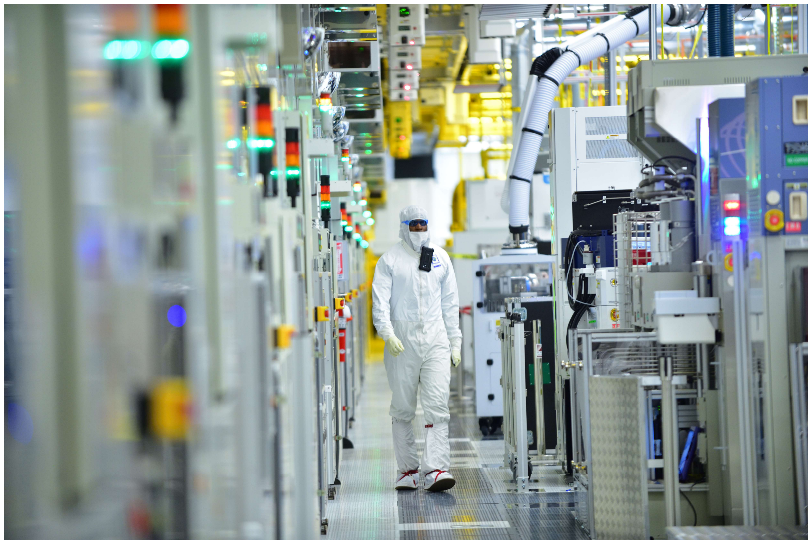


Fig. 1. Semiconductor fabrication plant (fab) environment (source: *Intel Corporation*).

The necessity of developing software solutions for robot simulation and analysis in the field of semiconductor device manufacturing is imposed not only by the uncompromising dynamics of this particular industry, characterized by its complexity, rapidly changing processes, cleanliness, flexibility, constant strive for superior quality, optimization and reduced costs, but also by the lack of currently available systems for motion simulation and offline programming that are compliant with the specifics of substrate handling robots and semiconductor industry automation in general. Therefore, a SolidWorks API-based simulation tool [4], proposed by the same author, has been implemented and efficiently utilized for simulation, offline programming, design, analysis, optimization and marketing purposes. Despite its multiple beneficial features and wide applicability in industry, the developed solution had a number of drawbacks, associated with its open source VBA macro-based nature, stability, reliability, 3D model complexity and others. In order to eliminate these disadvantages and to improve its functionality, the simulator had to be re-implemented in the form of SolidWorks add-in. The concept, architecture, advantages, and applicability of the newly-developed add-in version are thoroughly reviewed and presented in the paper's following chapters.

2. Development concept and architecture

SolidWorks add-ins are in-process applications for SolidWorks that enable high level of customization, optimal performance and user experience, using the SolidWorks API (Applications Programming Interface) [5]. Based on the COM (Component Object

Model) standard, the SolidWorks add-ins are most commonly created through the Visual Studio platform, using any COM-compatible language, such as C++, C#, VB.NET, and VB6 [6]. Some of the advantages of developing custom applications in the form of add-ins (compared to macros and stand-alones) are associated with faster code execution, full SolidWorks API utilization, direct availability in SolidWorks, more integrated approach, improved reliability, etc. [7, 8].

In addition to the already mentioned benefits, there are two main reasons in particular, which have motivated the initial version of the simulator for substrate handling robots to be re-implemented in the form of SolidWorks add-in – prevention of accessing the source code, and development of an efficient collision detection tool. Unlike macro-based custom applications, where the source code (contained in a swp-format file) can be relatively easily accessed, the add-in approach, which compiles DLL (Dynamic Link Library) files, provides much higher levels of security – quite an important feature, especially if dealing with commercially-oriented applications. On the other hand, the availability of an efficient instrument for preventing collisions in advance is a functionality with a great importance to the majority of industrial robotic applications, particularly to the field of semiconductor device manufacturing, where the combination of complex equipment, highly-restricted working volumes, bulky objects for handling, limited access, poor visibility and uncompromising throughput requirements increases the chance for potential collisions. At the same time, even minor interferences can be critical not only to the transported objects, but also to the automated tool and the whole manufacturing process (destroying a silicon substrate that is extremely fragile inside a cleanroom environment takes quite a lot of time and resources to clean up). It is therefore essential that the basic visual approach for collision detection of the macro-based simulator is substituted with a much more precise and effective tool for real-time interference monitoring.

The architecture of the implemented SolidWorks add-in for motion simulation, layout analysis and collision detection of substrate handling robots is based on the following main subroutines, shown in Fig. 2:

- *Declaration of parameters* – introduces all variables, constants, offsets, operators and interfaces, necessary for the add-in's functionality;
- *Declaration of functions* – formulates mathematical expressions used for various calculations in the different subroutines;
- *Main procedure* – defines the main SolidWorks API interfaces, fundamental for the add-in, as well as the constant Denavit-Hartenberg (D-H) parameters of the simulated robot (robot geometry);
- *Kinematic modeling* – features all kinematic calculations, associated with the simulated robot – variable D-H parameters, position vectors, rotation matrices, 4x4 transformation matrices, inverse kinematics, etc.; sets the algorithms for robot 3D model actuation and object-handling simulation;
- *Collision detection* – defines the logics for the implementation of the collision detection tool;

- *Connection with robot motion control software (MCS)* – a key subroutine, responsible for the actuation of the robot's 3D model in the SolidWorks environment; establishes a real-time connection (referred to as a 'Motion Loop') between SolidWorks and the robot MCS, thus allowing the application to read the current values of all robot axes (variable D-H parameters), required for simulating the robot's motion;
- *Graphical user interface (GUI)* – provides access to all functions of the application and serves as a link between the different subroutines, described above;

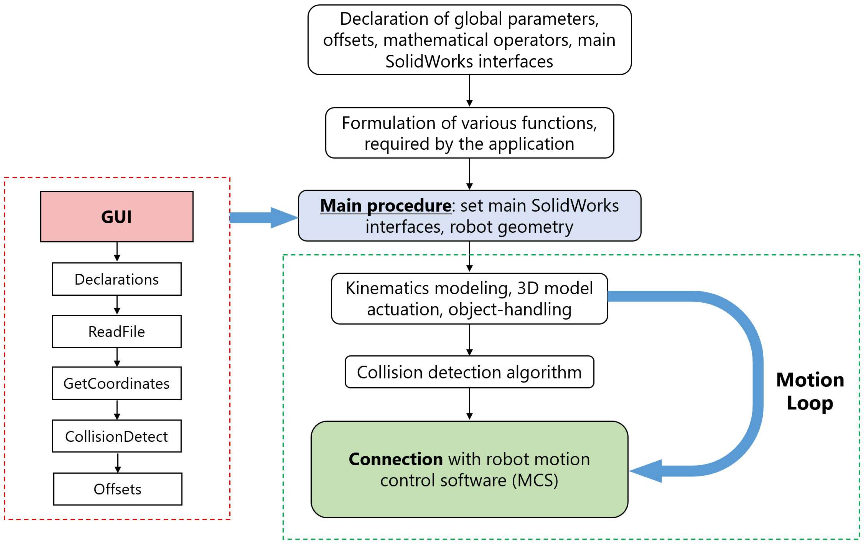


Fig. 2. Architecture of the SolidWorks add-in for simulation of substrate-handling robots.

In order to successfully utilize the developed add-in for simulation, layout analysis and collision detection tasks, each of the above subroutines has to be implemented in accordance with the characteristics of the specific substrate handling robot, which is going to be simulated. This is particularly valid for the *Kinematic modeling* subroutine, which provides essential information related to the structure and mobility of the specific robot – geometry, kinematics, number and type of joints, logical axes, units of measure, etc. It is therefore crucial for the add-in's functionality, accuracy and efficiency that precise kinematic models of the whole range of substrate handling robots are developed.

3. Kinematic modeling of substrate handling robots

Despite the relatively large diversity of geometries, arm configurations and sizes, most of the substrate handling robots are based on the so-called 'SCARA-type' structure (Fig. 3), as substrates in semiconductor manufacturing are typically handled in horizontal planes at different vertical positions within the reach of a human arm, often in a cylindrical coordinate frame [9]. The most significant difference from the standard SCARA structure is related to the relocation of the vertical axis from the robot base to the arm tip, imposed by the requirements for horizontal and thin end-effector that allows reaching into small openings, as well as for cleanliness considerations [10].

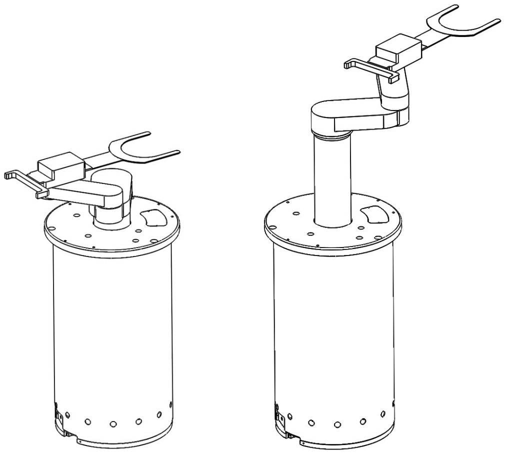


Fig. 3. ‚SCARA-type' – the most popular structure among substrate handling robotics.

The 'SCARA-type' structure is based on a serial open-loop kinematic chain comprised of several links (two or three), connected together by fifth-order revolute joints. For three-link arms, the first link always remains parallel to the third one, which is a specific feature of the mechanical design. The arm, operating in a cylindrical coordinate system and actuated by two or three motors can extend and retract in its plane of motion. It can also swing around its central axis, thus accessing all points within a planar section of the workspace. Another specific feature of this type of robots is that the user does not have direct access to the motion of each motor or joint. Instead, the so-called logical axes (described in Table 1), which correspond to one or more actuator's synchronized motion can be controlled (Fig. 4).

Table 1. Description of the logical axes of a typical substrate handling robot.

Notation	Unit	Name	Type	Description
T	0.01°	Theta	rotary	Rotation of the arm around the central axis
R	0.001″	R	linear	Extension and retraction of the arm in the plane of motion
Y	0.01°	Yaw	rotary	Rotation of the end-effector around its axis
Z	0.001″	Z	linear	Ascent and descent of the platform in vertical direction

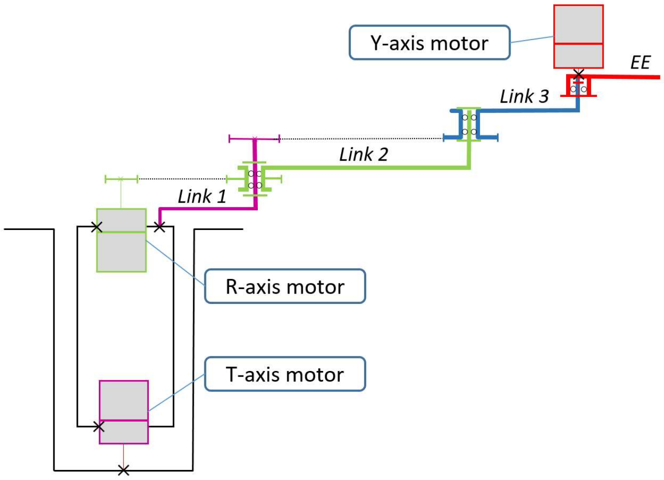


Fig. 4. Actuators arrangement and mechanical couplings of a 3-link substrate handling robot arm.

Observing the specific mechanical and actuator relationships, the final kinematic model of the three-link serial arm, typical for the majority of substrate handling robots, is derived by using the Denavit-Hartenberg approach (Fig. 5). The corresponding D-H parameters, necessary for the implementation of the SolidWorks add-in's kinematic subroutine, described in Chapter 2, are determined and presented in Table 2.

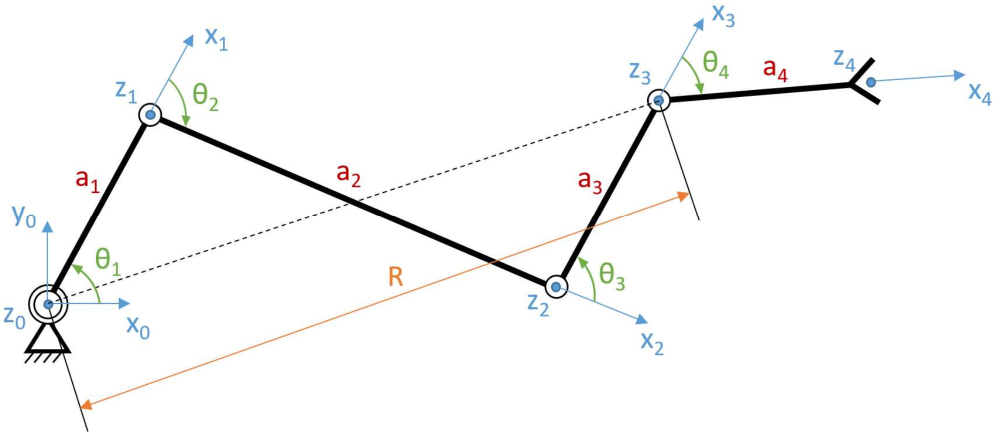


Fig. 5. Kinematic model of a 3-link substrate handling robot arm based on the D-H convention.

Table 2. D-H parameters of the 3-link arm, observing its mechanical and actuator relationships.

Link *i*	**DH Parameters**			
	ai *[mm]*	**α**i *[deg]*	**d**i *[mm]*	**θ**i *[deg]*
1	$a_1 = l$	0	0	$\theta_1 = q_T + q_R$
2	$a_2 = 2l$	0	0	$\theta_2 = -2q_R$
3	$a_3 = l$	0	0	$\theta_3 = 2q_R$
4	$a_4 = l_{ee}$	0	0	$\theta_4 = q_Y$

where

$\boldsymbol{q_T}$ = T – angle of rotation associated with the Theta logical axis

$\boldsymbol{q_R} = \sin^{-1}\left(\frac{\mathrm{R}}{a_1+a_2+a_3}\right)$ – angle of rotation associated with the R logical axis

$\boldsymbol{q_Y}$ = Y – angle of rotation associated with the Yaw logical axis

In addition to the serial 'SCARA-type' planar arm, a large number of the substrate handling robots, operating in fabs all around the world feature a special type of overconstrained parallel mechanism named GPR$^{\mathrm{TM}}$, developed and manufactured by Genmark Automation in 1996 [11]. The GPR mechanism has three degrees of freedom and is designed to perform two small independent rotations in the range of ± 2 degrees and a large-range translation (up to 20 inches). The terminal link (platform) is used as a basis for installing the serial 'SCARA-type' arm, resulting in a hybrid parallel-serial overall structure (Fig. 6), which is capable of adapting to misaligned equipment and compensating for the deflection of the manipulated objects [12].

Fig. 6. Hybrid parallel-serial structure of substrate handling robots.

The proper functionality of the SolidWorks add-in for motion simulation of substrate handling robots requires that kinematic models of the GPR mechanism must also be developed. Due to its overconstrained parallel nature, the GPR modeling procedure is much more complex compared to the serial ‘SCARA-type’ arm, as methods from the analytical mechanics are used in order to find the relationship between the finite rotations of the platform and the inherent imperfections of the GPR components, responsible for the three-degree-of-freedom mobility of the actual manipulator. Because of the limitations imposed on the size of the paper, the GPR kinematic modeling, which has been thoroughly explained in some of the author’s previous works [13], is not going to be presented here.

4. Functionality and applicability of the developed SolidWorks add-in

Once precise kinematic models of the hybrid parallel-serial structure of substrate handling robots are developed and properly implemented into the source code of the SolidWorks add-in simulator, the last can be efficiently utilized for various beneficial activities, including realistic motion simulations, offline programming, layout analysis, collision detection, throughput estimation, concept validation, performance optimization, marketing and demonstration tasks. In order to demonstrate the capabilities of the proposed development, a basic automated layout, comprised of the GB7Y GPR substrate handling robot, designed and manufactured by Genmark Automation, a cassette for storing 150 mm wafers, and a device, called pre-aligner, utilized for centering and orienting the handled objects, has been chosen. The actual physical system and its corresponding SolidWorks 3D assembly can be seen in Fig. 7. The purpose of the experimental study is to demonstrate the functionality of the developed SolidWorks add-in-based simulator for substrate handling robots, and to validate its practical applicability. The process consists of two main tasks – station teaching of the proposed layout (where the cassette and the pre-aligner are located), and execution of interference analysis in order to verify that the taught stations and programmed robot trajectories are optimal and collision-free.

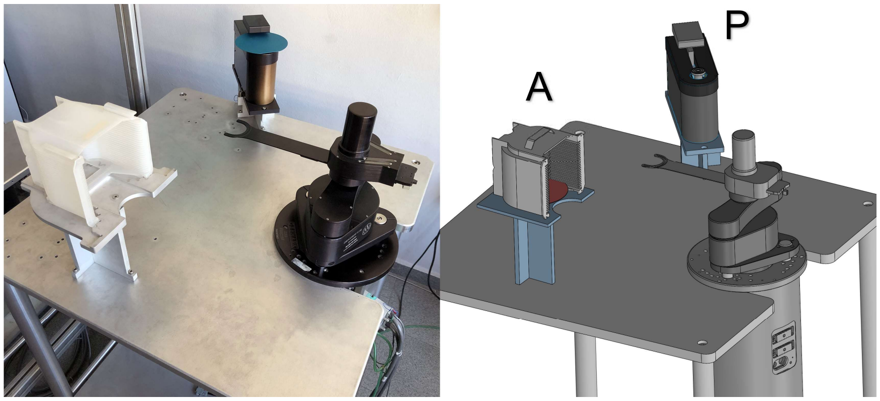


Fig. 7. The actual substrate handling robot layout (left) and its 3D model counterpart (right).

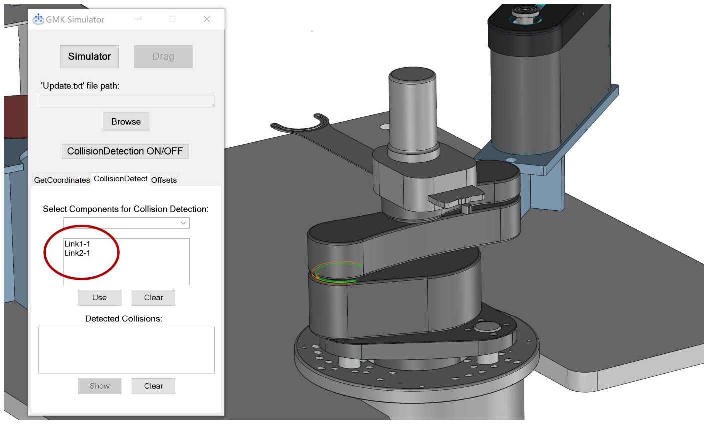
The purpose of the first task is to demonstrate the capabilities of the developed SolidWorks add-in simulator in the field of station teaching. This is a fundamental procedure, essential for the set-up, reliability, efficiency and throughput of the actual substrate handling robots, especially when complex tools, comprised of hundreds of different stations have to be programmed. In such cases, the availability of a virtual offline programming tool that would allow the teaching process to be preliminary and remotely executed is of a great importance to nowadays semiconductor industry automation. Despite the simplicity of the analyzed layout, given in Fig. 7 (comprised of only two stations A & P), it clearly demonstrates the performance of the offline teaching procedure. It is based on the 'GetCoordinates' functionality of the simulator that allows the user to manually drag the robot arm around the environment and derive the robot's coordinates for the desired position (station) that has to be taught. In this manner, the coordinates of the two stations A and P are derived (Table 3), and efficiently utilized in the teaching of the actual physical equipment.

Table 3. Station coordinates of the analyzed layout (derived offline via the SolidWorks add-in).

Station	**Robot coordinates (logical axes)**			
	T	***R***	***Y***	***Z***
A	– 6392	13621	932	1600
P	4490	10515	– 3484	2269

The second task of the experimental study seeks to demonstrate the add-in's collision detection tool, which allows the user to perform real-time interference analysis in a safe virtual environment, thus preventing damage to the physical robot and to the whole automated system. The tool, entitled CollisionDetect, is extremely useful not only for inspecting potential 'external' collisions between the robot arm and the static tool equipment (a process that can in general be visually performed), but also to determine if any of the robot's hard stops will eventually reach its limits during the execution of the desired motion. Namely the second case, which is rather important and more difficult for application engineers to foresee in the course of analyzing the specific customer's layout, is illustrated in this section.

As both stations A and P have already been taught, the user can proceed with the execution of the interference analysis by selecting the desired components, which the system is supposed to monitor for potential collisions. A possible approach, associated with checking the R-axis range of motion requires that the robot arm's first and second link, where the R-axis hard stop itself is implemented in the form of a pin moving inside a radial slot, are selected (Fig. 8).



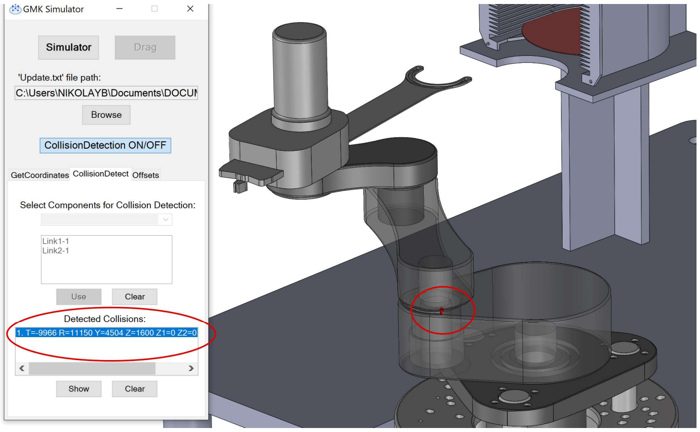
'Update.txt' file path:

Select Components for Collision Detection:
Link1-1
Link2-1

Detected Collisions:

Fig. 8. Interference analysis set-up procedure associated with monitoring the R-axis hard stop.

The next step is to execute the desired motion cycle, connecting the two stations A and P, and analyze the results of the collision detection check. As it turned out, approaching station A with the arm configuration, specified in Table 3 is not going to be possible, as one of the R-axis hard stop limit is eventually reached during the process (Fig. 9). Based on the information, provided by the add-in, the user can observe where exactly the collision takes places and take measures for avoiding it with the actual system (e.g. selecting another arm configuration for attaining station A).



'Update.txt' file path: C:\Users\NIKOLAYB\Documents\DOCU

Select Components for Collision Detection:
Link1-1
Link2-1

Detected Collisions:
1. T=-9966 R=11150 Y=4504 Z=1600 Z1=0 Z2=0

Fig. 9. Interference analysis set-up procedure associated with monitoring the R-axis hard stop.

5. Conclusion and future work

The development, implementation utilization and applicability of a SolidWorks add-in for motion simulation, layout analysis and collision detection of substrate handling robots is presented in the current paper. The efficiency of the proposed tool has been validated with the help of a typical substrate handling robot (GB7Y) and a basic layout with two main stations. The physical equipment has been successfully analyzed and set-up by utilizing some of the simulator's main functions related to realistic motion simulations, offline programming, station teaching, collision detection, etc. The future work associated with the project is mainly oriented towards developing a SolidWorks-independent version of the proposed system.

ACKNOWLEDGEMENTS

This work is supported by the Technical University of Sofia, Bulgaria under Research Project №201PR0003-08/2020.

REFERENCES

[1] D. Liao, "Automation and Integration in Semiconductor Manufacturing," Semiconductor Technologies, Chapter 3, pp. 39-59, April 2010.

[2] M. Hosek, J. T. Moura, "Advanced Control Techniques for Semiconductor and Flat Panel Display Substrate Handling Robots," Proceedings of the 17th World Congress, The International Federation of Automatic Control, pp. 6731-6738, July 2008.

[3] N. Bratovanov, "Modeling, Simulation and Offline Programming of a Class of Robots, Applicable to Semiconductor Industry Automation," Ph.D. Thesis, Sofia, Bulgaria, 2019.

[4] N. Bratovanov, "Robot Modeling, Motion Simulation and Off-line Programming Based on SolidWorks API," Third IEEE International Conference on Robotic Computing (IRC), vol.1, pp. 574-579, 2019.

[5] "How to develop add-ins for SolidWorks automation via API," Extracted on July 6, 2020, URL: https://www.codestack.net/solidworks-api/getting-started/add-ins/

[6] S. Riyaz, "What's the Difference between SolidWorks Macros, Add-Ins and Executables?," Extracted on July 6, 2020, URL: https://www.javelin-tech.com/blog/2020/04/solidworks-macros-add-ins-and-executables/

[7] K. Rice, "Macros vs Add-ins vs Stand-Alones," Extracted on July 6, 2020, URL: https://www.cadsharp.com/blog/macro-vs-add-in-vs-stand-alone/

[8] L. Malpass, "SolidWorks 2009 API. Advanced Product Development," AngelSix, 2008.

[9] Y. He, J. Chen, J. Gao, C. Cui, Z. Yang, X. Chen, Y. Chen, K. Zhang, H. Tang, "Research on Motion Simulation of Wafer Handling Robot Based on SCARA," 19th International Conference on Electronic Packaging Technology, pp. 734-739, August 8-11, 2018.

[10] K. Mathia, "Robotics for Electronics Manufacturing Principles and Applications in Cleanroom Automation," Cambridge University Press, 2010.

[11] G. Genov, D. Todorov, "Universally Tiltable Z-Axis Drive Arm," US Patent US5954840, June 13, 1996.

[12] G. Genov, Z. Sotirov, E. Bonev, "Robot Motion Compensation System," US Patent US6489741B1, August 1998.

[13] N. Bratovanov, Z. Sotirov, V. Zamanov, "Mobility and Accuracy Analysis of a Class of Overconstrained Parallel Mechanisms," Mechanics of Machines, ISSN 0861-9727, Year XXVI, issue 2, pp. 26–32, 2018.

Author: Nikolay Bratovanov, Assistant Prof., PhD, Faculty of Automatics, Technical University of Sofia, e-mail: nbratovanov@tu-sofia.bg