



HAL
open science

Generation of safe and intelligent tool-paths for multi-axis machine-tools in a dynamic 2D virtual environment

Rafiq Ahmad, Stéphane Tichadou, Jean-Yves Hascoët

► **To cite this version:**

Rafiq Ahmad, Stéphane Tichadou, Jean-Yves Hascoët. Generation of safe and intelligent tool-paths for multi-axis machine-tools in a dynamic 2D virtual environment. *International Journal of Computer Integrated Manufacturing*, 2016, 29 (9), pp.982-995. 10.1080/0951192X.2015.1130258 . hal-01265764

HAL Id: hal-01265764

<https://hal.science/hal-01265764v1>

Submitted on 21 Nov 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - NonCommercial 4.0 International License

Generation of safe and intelligent tool-paths for multi-axis machine-tools in a dynamic 2D virtual environment

Rafiq Ahmad^{a,b,*}, Stephane Tichadou^c and Jean-Yves Hascoet^d

^aMechanical Engineering Department, CECOS University, Peshawar, Pakistan; ^bResearch Unit of Engineering sciences (RUEs), FSTC, University of Luxembourg, Luxembourg, Luxembourg; ^cSYMME University of Savoie, Annecy, F74000, France; ^dIRCCyN, Ecole Centrale Nantes, Nantes, France

Complexity of multi-axis, multi-functional computer numeric control machine tools and high productivity requirements with high precision is increasing day by day. This enhances the growing importance of machine intelligence, automation and safety. Integration of different types of manufacturing process, for example, Mill-Turn is the need of modern technology but this integration also increases the profile of collision risks. The general context of this study is the multi-axis production process automation and intelligence (to work and react automatically) for safe tool movements. The work done during this research should appear in the form of machine intelligence for automatic safe and efficient collision-free tool trajectory generation without collisions. This study focuses on development of an intelligent approach for enhancing decision processes by providing automatic solutions to the collision problem. The main objective is to make machine vision control effective, that is, images or video processing, so that it can see activities regarding collisions and can react or command automatically for safe production. In the presence of obstacles, the proposed approach will provide decisions regarding trajectory correction and improvement. The proposed vision tool is able to take into account the evolution of the scene, such as the aspects of obstacle changes (shape, size or presence) during production. Results presented are based on automatic 2D traversal safe trajectory generation and correction in a virtual simulated but dynamic environment. The approach presented will be adapted to online collision problems on the shop-floor by integrating it with step-compliant data interface for numeric controls technology in the future.

Keywords: multi-axis CNC machines; tool-paths; safe trajectory generation; collision detection and avoidance; image processing; machine intelligence

Introduction

The demand of high speed and precise production is increasing with the increasing effect of complexity of production parts and processes. This ultimately increases the demand of automatic decision process during online (real production) and preparation (virtual simulations). Multi-axis machines are known for rapid production using multi-tool working on the same work-pieces simultaneously in order to achieve precision and time efficiency with high customer satisfaction. It is therefore, important to deal with many objects moving in the same machining environment. These moving objects could be many tools working on the same or different work-piece in the same environment or it could be a production tool and some moving machine parts. Complex virtual and real multi-axis machine-tool examples are shown in Figure 1.

In order to deal with tool movements in a dynamic scene two important aspects need to be taken into account: (1) dynamics of the scene (for control and collision verification), for example, respective speeds (feed rates), accelerations/deceleration and (2) objects missions in the scene (for prioritising object trajectories)

It is important to deal with dynamic environments where multiple tools are working because a static environment is not always the case in multi-axis machining except in simple production processes such as 2.5D milling and drilling operations. Conventional complex technologies in multi-axis machines are widely explained by Moriwaki (2008). Multi-tool working in the same environment increases machining efficiency and minimises time to market but, at the same time, increases machining complexity and hence problems of collision.

An efficient approach is required to deal with such collisions to generate safe trajectory especially for non-functional trajectories where relatively high speeds are concerned (Hascoet, Bennis, and Guerin 1993). So far, a lot of work has been done for functional trajectories in the context of multi-axis machine-tools (Makhanov 2007a, 2007b; Bohez et al. 2003; Tang, Bohez, and Koomsap 2007) but, despite the high speed of cutting tools during non-functional trajectories, as compared to functional trajectories, less attention has been paid to analyse such types of trajectories. A few manual solutions exist, or else the process of collision avoidance in non-functional trajectory

*Corresponding author. Email: rafiq.ahmad@uni.lu

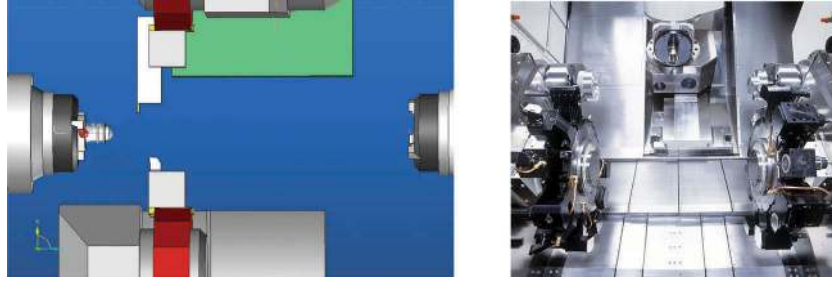


Figure 1. Examples of machines with multi-tool working on the same work-piece; (left) a virtual machine example (Index G300); (right) a real machine example (Nakamura).

is the operator's responsibility, therefore, human intervention is vital. Hence, conventional computer-aided manufacturing (CAM) software and computer numeric control machines can detect possible collisions but leave the avoidance decisions to the operator. In modern CAM software such as TopSolid (Messler), a function can be activated for collision verification, which can detect collisions but does not provide any automatic solution for their avoidance.

Safe trajectory generation and collision avoidance is an important topic of research for different technological areas (Jiménez, Thomas, and Torras 2001). Lee and Kardaras (1997) and Petiot, Chedmail, and Hascoet (1998) worked on the concept of elastic strings with the integration of the idea of potential fields that give good results for trajectory generation but are mostly useful for robotic offline trajectory generation. A generalised pattern search algorithm (Ata and Myo 2006) is presented to generate a collision-free trajectory for a planar redundant manipulator. Wang, Li, and Zhou (2006), Morishige, Kase, and Takeuchi (1997) and Jun, Cha, and Lee (2003) focused on collision detection in multi-axis machines, where visibility data using computer vision and a C -space approach were the methodologies in focus, respectively. Visibility data for collision detection is a good approach for extracting real information of the scene for collision detection but needs further improvements for automatic collision avoidance. The C -space approach in its present form for collision detection works on 2D information, and, therefore, 3D data extraction from the scene and rapid displacement (non-functional) trajectory generation are the primary concerns in order to use this method for machine-tool collision avoidance. An algorithm for interference analysis between a manually predefined tool and an arbitrary work-piece was also developed by Ho, Sarma, and Adachi (2001). One of the big problems addressed by Balasubramaniam et al. (2002) in 5-axis machines is the tool access, which is treated as a collision avoidance problem. A collision-free path based on D -plan algorithm for part disassembly simulation has also been treated (Zhang et al. 2008). The widely used A* method,

an extension of Edsger Dijkstra (Bennewitz and Burgard 2000), (Moravec and Elfes 1985), determines the cost optimal path for multi-robots. First, trajectories for each robot are calculated separately and then possible conflicts in trajectories of robots are checked.

Very few methods exist in the literature for collision detection and avoidance in a multi-axis machining environment especially when a dynamic environment is concerned. A vision-based method (Heidelberger, Teschner, and Gross 2004) was selected because of the many advantages as listed in our previous work (Ahmad, Tichadou, and Hascoet 2012), for example, capturing real information from the scene, low cost equipment, etc. Vision-based tool is used for safe trajectories generation in multi-tool dynamics environment. In this paper, for each scenario, where multi-tool work in the same environment, it is assumed that one tool trajectory is programmed at a time and all other moving or static objects are considered as obstacles. Therefore, each tool under consideration will be moving in a dynamic environment, where collision detection and avoidance will be treated with static and moving obstacles, that is, another tool in motion.

Collision detection and avoidance with static obstacles was already treated in our previous work (Ahmad, Tichadou, and Hascoet 2010, 2012, 2013), whereas safe tools-path generation in a dynamic virtual scene with multi-tool in motion is the original contribution of this paper. This has not been done previously in the literature in the context of multi-axis machine-tool. The approach presented takes into account machining information such as tool mission and respective speeds in the scene for an effective decision. The information obtained from the vision systems is also used to update the real information about the scenario on-hand.

This paper will therefore integrate the approach of expert system (Castro et al. 2011), which detects the presence of risk of collisions between objects from the video frame analysis with our previously developed vision-based methodology (Ahmad, Tichadou, and Hascoet 2013, 2012) and enhance the system capabilities for automatic collision avoidance in multi-axis machining scenarios in virtual environments as an initial application towards real manufacturing.

Virtual manufacturing (Dugas et al. 2002) should enhance people’s ability to predict potential problems (collisions) and inefficiencies reflecting reality in product functionality and manufacturability before real manufacturing occurs. The system INtelligent and Safe Trajectory AT Expert (INSTATE) production system presented in the next sections uses the principle of our previous developed methodology, that is, a Snakes and Ladders based Analogy for Production Trajectory (Ahmad, Tichadou, and Hascoet 2012), but is adapted to the dynamic scenarios where rapid collision detection and avoidance is required.

General structure of the INSTATE vision system for expert production

The INSTATE system general diagram for safe tool trajectories generation in a global and dynamic virtual or real environment is shown in Figure 2. The system takes video frames or image shots at a certain time instant t defined from the scene that ultimately captures the real movements and locations of objects. The system finally generates safe and intelligent trajectories for a virtual scene, while using manufacturing information such as velocities and interpreted trajectories in both real and virtual scenes.

This INSTATE system is based on machine vision, which may take/integrate information from/between both virtual (computer-aided design (CAD)-CAM) to/and real (machine) scenes in order to correct and synchronise various characteristics of a machine, that is, type, dimensions, speeds and object localisation as detailed below. The final safe trajectory is generated only for a virtual simulation environment as an initial step and will be implemented in a real machining scenario in future.

- (1) *Type*: Objects must be identified to understand their required missions in the scene which is

important for collision avoidance process, for example, the collision detection with a moving tool needs more attention compared to a static machine part.

- (2) *Dimensions*: The real dimensions of objects in the scene must be communicated from the real machine to the system because the difference between such data may cause collisions (Ahmad, Tichadou, and Hascoet 2013).
- (3) *Speeds (tool feed rates)*: The real speeds of production tools must be communicated back from the machine to a virtual simulation. A collision scenario is explained in the results section for explaining the notion of dynamics and impact of speed difference between virtual and real production.
- (4) *Object localisation*: The term localisation refers to ‘the exact object location information’ in the scene. It is also important that object localisation data be the same in virtual as well as real scenes for avoiding collisions.

It is easy to use cameras and back tracking of information in a virtual environment because of no accessory costs and clear background; therefore, this paper initially focuses on virtual simulations where automatic collision avoidance is impossible today.

INSTATE production system

INSTATE production system, which is the key idea of the paper performs:

- taking into account the context of the machine
- taking into account the information obtained from the CAM preparation, real machine and vision tool

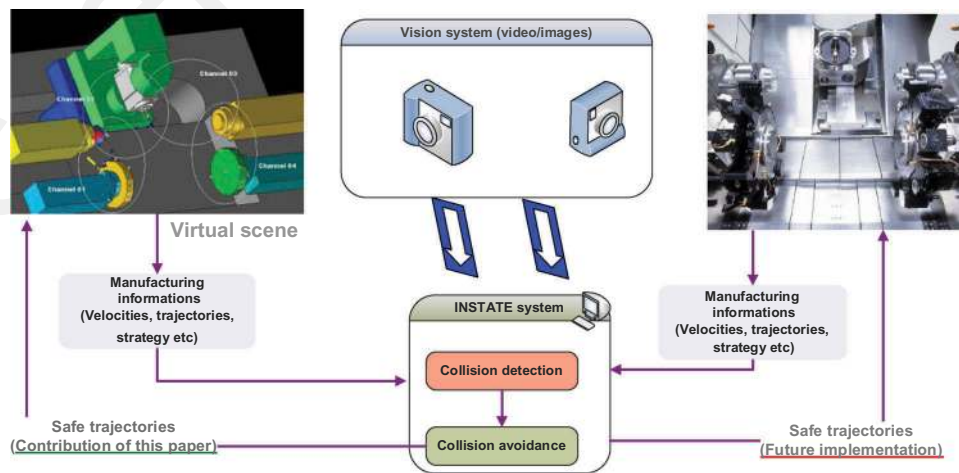


Figure 2. INSTATE system general diagram (Machines: Nakamura Super NTX).

- makes virtual production (real production→ in future) processes expert in the sense of detecting and avoiding collisions
- works without any human intervention (automatic)
- generates safe and intelligent trajectories in static and dynamic environment.

Figure 3 shows the general architectural diagram of the system where video/images are taken from machining scenes. These video images are processed to detect and identify objects present in the scene using Matrox Inspector software. The machine-tool speed data along with trajectory interpretation data with machine strategy (e.g. straight line movements) is transferred to the system from virtual CAD-CAM data. This data is treated by Translator 01 for converting the required information from machine to image frame of reference, that is, from ‘meters-per-sec’ to ‘pixels-per-sec’. All machining information along with video images at time instant t is transferred to the INSTATE algorithm module, used for collision detection and avoidance purposes.

The INSTATE algorithm module then detects collisions and takes some corrective measures to avoid such collisions intelligently. This trajectory correction process is done by accelerating/decelerating a tool or shifting a tool to a secure position and generating a safe trajectory

accordingly. The transfer of trajectory points to the machining environment is done by the Translator 02 which re-converts the safe trajectory coordinates from the image frame of reference to the machine frame of reference for application in the production simulation.

The virtual simulation process should produce trajectories according to norms and expectations of reality. The proposed vision-based INSTATE system promises automated and collision-free online tool trajectories. A video frame or fresh image at a certain time instant t is taken from the CAD-CAM virtual environment in order to get the actual situation and updates of the scenario for example as shown in Figure 4. The approach presented in this paper divides objects into two categories, that is, production tool and obstacles. Only one production tool is prioritised and its trajectory is generated and all the rest of the objects in the scene are considered as static or moving obstacles.

This time t will depend on many factors such as those given below:

- Security margin, ($\text{Dist}_{\text{safe}}$): This safe distance can be calculated for accumulative relative speed between tool and obstacle which is calculated by the formula: $\text{Dist}_{\text{safe}} = \bar{v}^2 / 2 \times \gamma$ (Sasiadek and Duleba 2000) (1)

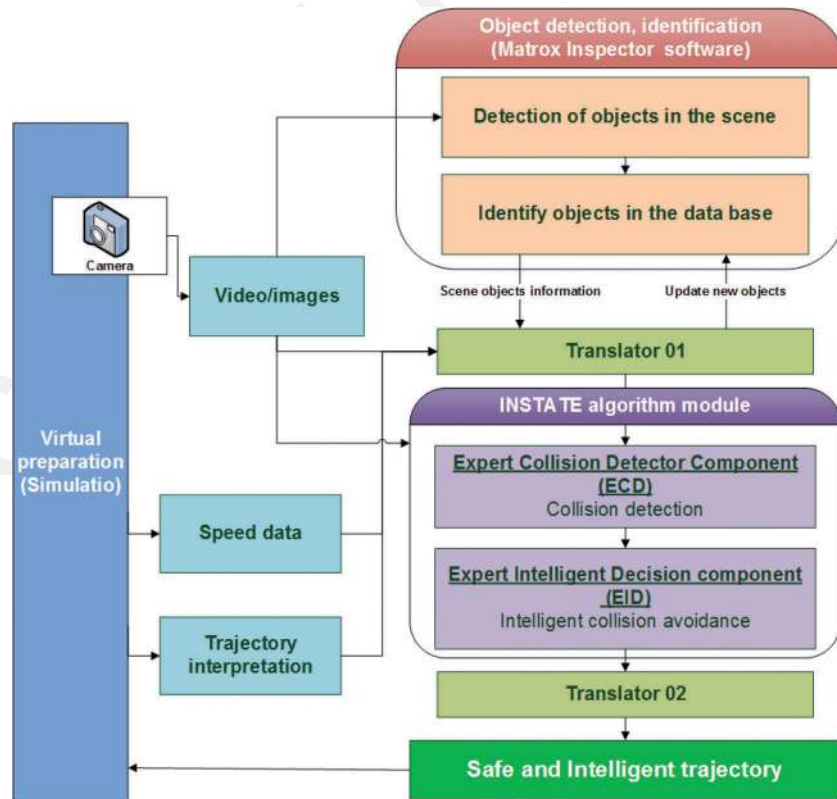


Figure 3. General INSTATE system architectural diagram for virtual production.

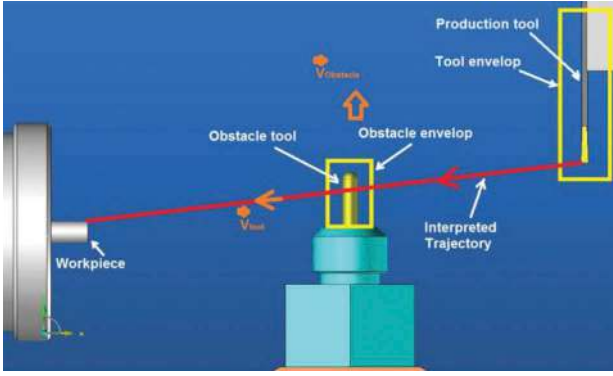


Figure 4. Scenario: Virtual collision scenario.

where, \vec{v} = relative speed between tool and obstacle;
 γ = acceleration.

- Production tool speed from CAM or tool tracking in images.
- Obstacle speed known from CAM or object tracking in images.

The time t is decided by the extra height/width of obstacle envelope/security margin (safe distance $\rightarrow \text{Dist}_{\text{safe}}$) to avoid collision at a distances as shown in Figure 5.

Note: The trajectory of obstacle is linear, that is, \vec{v} obstacle = constant. The application of INSTATE system architecture for the virtual production is shown in Figure 6, which gives a detailed view of how this system will work for collision detection and avoidance in virtual online simulation process. The INSTATE system extracts images (frames) at a time instant t from the video captured or by taking any fresh shot of the scene at the corresponding instant of time t chosen. These images or extracted

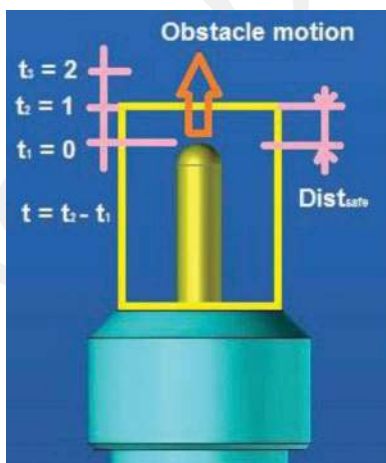


Figure 5. Calculating the time interval t : Obstacle motion is continuous between each time interval, whereas envelope jumps to the next position when time interval passes.

frames are pre-treated before applying the trajectory generation process.

The pre-treatment process is for differentiation of background from object elements in the scene in order to get the free space areas for safe tool movements. It converts colour images into binary images, where white colour (pixel value = 1) is assigned to objects and black colour (pixel value = 0) represents the free space or background area.

Images are treated one-by-one at their specific time instant by the algorithm module of the decision system presented in the next section. Each tool trajectory next point is verified for collision by locating it in each fresh image taken from the scene at the time instant. In the case of collision, the presented concept takes corrective action to avoid this collision intelligently by providing a secure position using a Matlab programming based algorithm. Figure 6 also shows some input data from the machining environment through Translator 01 to the algorithm module apart from the video images. This data include information about trajectory interpretation, machine strategy (straight line for simplicity), initial trajectory point (tool start point) and final trajectory point (production point), obstacle speed and positions at the time instant and machine production tool speed.

The system uses closed loop, that is, by taking video or sequence of images from a CAD/CAM scene and pre-processing them to be used by the algorithm module that in turn uses CAM data regarding the scene and finally generates safe trajectory points which are transferred back into the CAM simulation process. The video images are taken from the scene, where production tool and obstacle are moving with their corresponding missions and their initial and final points known from CAM or objects detection in the scene.

Several time spans are selected to take image frames from the video or fresh images of the scene are taken and are processed in the pre-treatment block as shown in Figure 6. These images are transferred into the INSTATE algorithm module at each time instant t . The INSTATE algorithm module also takes necessary information from the CAM data and real machine data, which are before converted from the machine coordinates frame of reference to image coordinate in the Translator block 01, if necessary. The safe trajectory points generated at each instant are transferred to the machine coordinates system by Translator block 02. The final corrected tool trajectory and machine-tools known velocity calculates the abscissa curvilinear and time at each trajectory point. Finally, all the machine coordinates data are fed-back into the CAM simulation process in order to use the safe and intelligent trajectory generated with corresponding tool speeds.

This process will generate a valid and intelligent solution for collision detection and avoidance in a virtual production process for a dynamic scene with simple 2D transversal non-functional trajectories (rapid transverse).

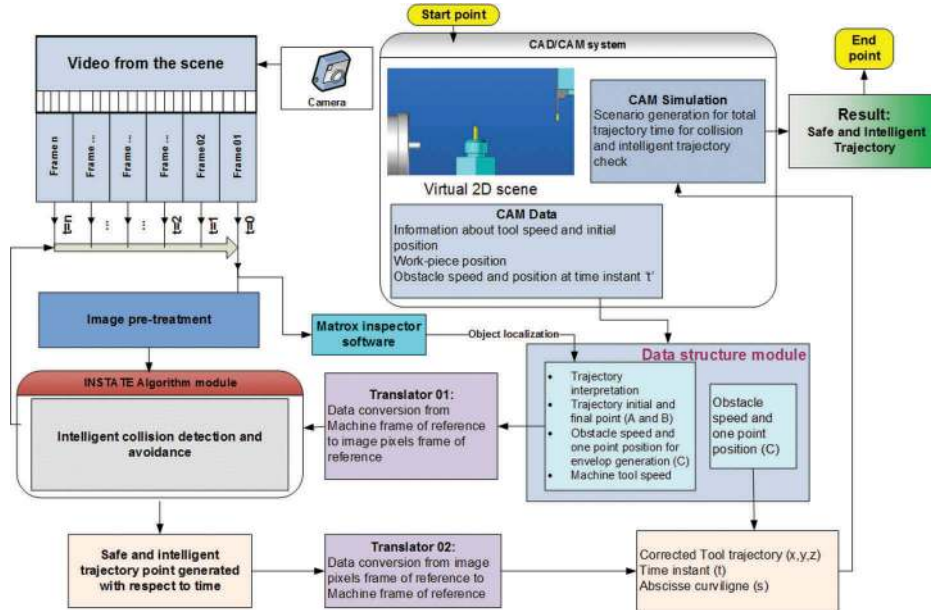


Figure 6. INSTATE system architecture diagram for virtual production.

Important components of the system:

- **Translator modules:** Used for 2D-to-3D and 3D-to-2D conversion. This translator is used to convert data from the machine frame of reference to the image frame of reference and vice versa. The translator will receive as input the outcomes of the knowledge extraction system about object detection, identification and 2D tracking from video images. The translator module is responsible for converting input events, object information and trajectory data into pixels frame of reference and vice versa.
- **Processing Unit or Algorithm module:** INSTATE algorithm is the processing unit. This unit is further divided to two units: expert collision detection component (ECDC) and expert intelligent decision component (EIDC).

Collision detection and avoidance is done by generating safe trajectory according to the machine strategy while using manufacturing data and verifying each next trajectory point.

INSTATE production algorithm module

The detailed description of the INSTATE production algorithm module following the procedure is given below.

INSTATE algorithm module description

The INSTATE algorithm module presented can be divided into many components as shown in Figure 7. In Level 1, two major components of the INSTATE algorithm module are shown that basically work for the two important steps of safe trajectory generation process, that is, collision

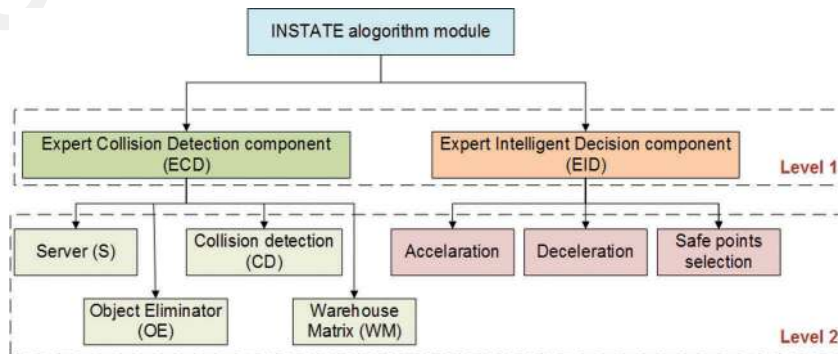


Figure 7. INSTATE system components level diagram.

detection and collision avoidance. These two components are further detailed on Level 2, which provide further in depth information about the two components.

The two major components are Expert Collision Detection (ECD) and Expert Intelligent Decision (EID), which use the information regarding scene and manufacturing data for generating safe trajectories according to the machine strategy. The ECDC detects the collision risks while the EIDC provides assistance for its avoidance. The two levels presented are just for understanding the level of information flow. The two components shown on Level 1 and their detailed internal modules on Level 2 are explained further below.

Expert collision detection component

The ECDC has the responsibility of detecting collisions while verifying each trajectory point during tool motion according to machine strategy. The ECD is a component of the INSTATE algorithm that detects expected collisions as shown with an example in Figure 8. In the case of collisions, an alert call is made into the EIDC for

intelligently/automatically avoiding these collisions by proposing an expert solution.

The ECDC is further divided into many sub-modules as per the definition and functionalities given in Figure 9 and in the text below:

Server (S). Responsible for receiving the events from Translator, Matrox inspector software (for example) and storing image information using pixel data extraction. It stores all the information about objects in the scene, for example, their speed, status, mission and tracking records. It also provides a new extracted pixel data matrix to the 'Scenario Object Warehouse (SOW)' by taking information from fresh image frame at the corresponding time instants for updating the scene information.

Scenario Object Warehouse (SOW). It shows the different objects that exist in the scenario in real-time at the corresponding time instant. All previous knowledge and new knowledge generated by the overall system is stored and integrated in the warehouse.

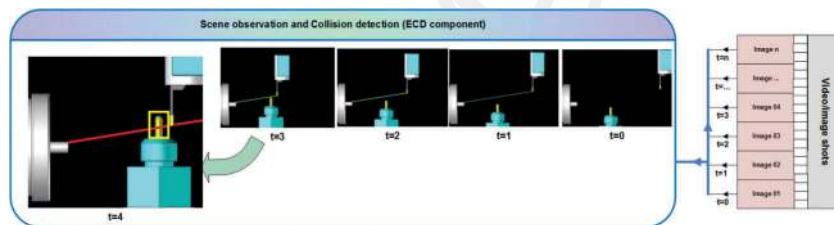


Figure 8. ECDC functionality for a virtual multi-axis machine scenario.

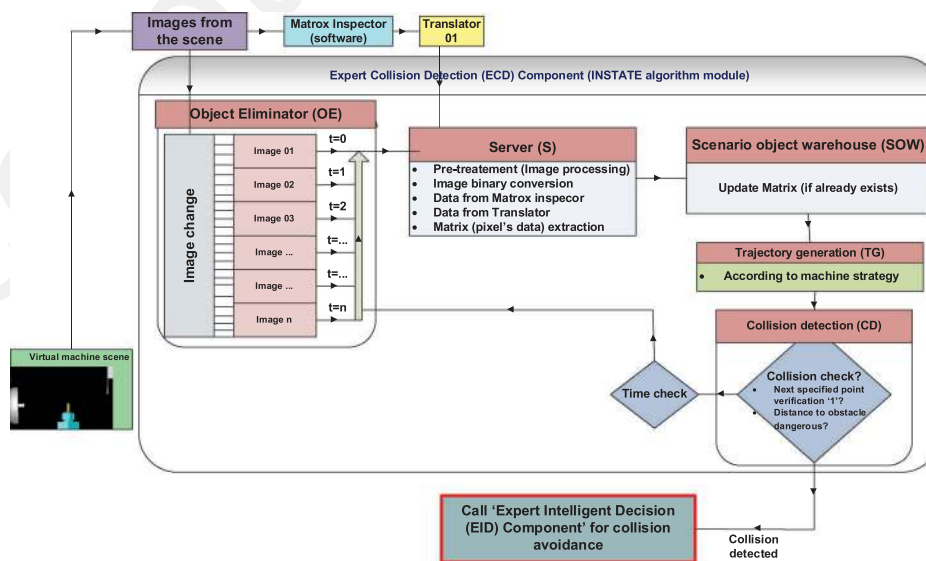


Figure 9. ECD sub-components.

Trajectory Generation (TG). It takes all the information regarding current scenario in the scene from data structure, that is, tool strategy, mission, speeds, etc. and finds the next trajectory point according to the machine strategy.

Collision Detection (CD). The CD then verifies the next trajectory point obtained from the TG module for collision. If the pre-defined collision value occurs at the next trajectory point it means collision is expected and the CD calls EIDC for further decisions.

Object Eliminator (OE). Every time when a collision is detected, the time interval passed is verified for the new fresh image from the video or the scene. OE is called for image change that checks the time instant t for each frame (or image fresh shot) change for processing. The OE changes the input image to the Server module at a certain time instant t .

Once collision is detected at a certain point in a trajectory, the EIDC is called for taking intelligent decisions for real-time collision avoidance. The EIDC for taking intelligent decisions according to the scenario on-hand for collisions avoidance is discussed next.

Expert intelligent decision component

In the case of a collision, the ECDC calls the EIDC for intelligent trajectory collision avoidance decisions. The EIDC takes further action by looking into the scenario of the problem. The EIDC can take a few major possible decisions for the production tool according to a list as prioritised by numbers.

- (1) Accelerate tool speed in order to avoid falling in the obstacle path and hence avoiding collision in real-time,
- (2) Decelerate tool speed in order to leave free space to the obstacle,
- (3) Detect a safe point (obstacle corner/space point) and divert tool motion towards it for safe trajectory generation,
- (4) Stop tool motion and generate alarms so that the operator takes some manual safety precautions/actions to avoid the expected collision. This manual action could be: changing the process plan or the machine; the sequence/order of operations or the process itself. This solution is not treated here and will be treated elsewhere.

It is assumed in this paper that the obstacle motion is constant and will not be disturbed for collision avoidance; therefore, the implementation process will only focus on the third solution, that is, avoiding tool collision by properly selecting a secure position. The EIDC is therefore divided into the following three modules according to the decisions discussed above. These modules are also highlighted in Figure 10.

Acceleration. This sub-component has the responsibility for checking the distance to the obstacle (Security Margin, Equation (1)) and at each trajectory verification point and taking a decision to accelerate the tool accordingly.

Note: The velocities of obstacle and tool can be obtained from the CAM data and image analysis by tracking objects in the scene. The acceleration data is not available in the CAM process, and therefore its value can only be obtained from operator experience.

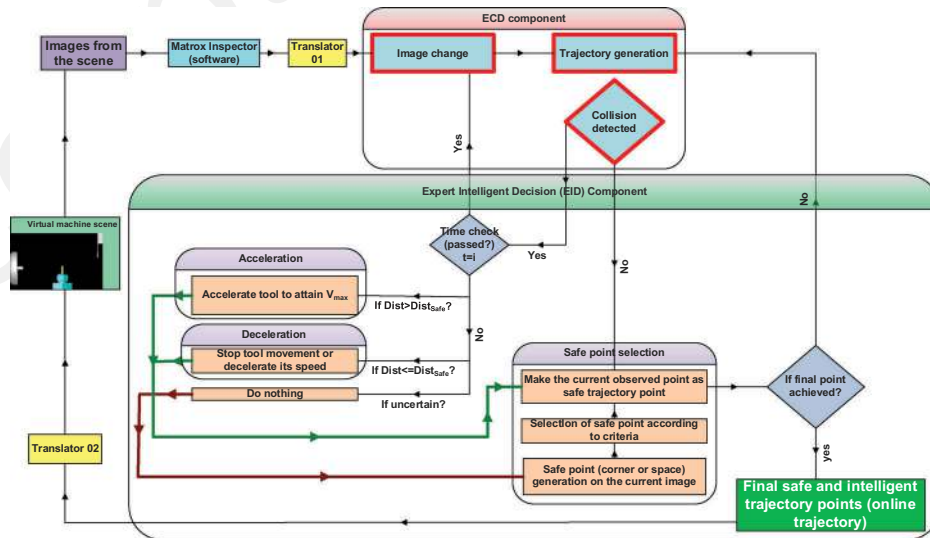


Figure 10. EIDC sub-components.

Therefore, for the current module: if Actual distance $> \text{Dist}_{\text{safe}} \rightarrow$ accelerate tool to attain V_{max} , where V_{max} = maximum machine tool speed achievable for the specific machine. Velocity management can be done as proposed by Sasiadek and Duleba (2000). According to the authors, the vehicle compares its current velocity with a velocity considered safe in the given environment. If the difference is negative, it increases the current velocity, otherwise it decreases the velocity.

Deceleration. Deceleration of tool speeds or stopping the tool movement in the machining scene is the case where the system is unable to find a solution because of scene congestion or if it is not possible to pass the obstacle even with maximum attainable velocity. This module decelerates the tool and leaves the trajectory way to the obstacle and giving priority to the obstacle to pass before.

Therefore, for the deceleration module: if Actual distance $< \text{Dist}_{\text{safe}} \rightarrow$ stop tool movement or decelerate its speed

Safe point selection. In the case where the acceleration and deceleration modules are unable to avoid collision then safe point selection module is activated for this purpose.

This module generates a safe trajectory while using the safe points in space as shown in Figure 11.

The following major steps are used by the EIDC to find safe points for tool diversion:

- *Safe corner detection:* Safe corners are detected for the input image binary matrix as per the procedure given in Figure 12.
- *Safe space points detection:* Safe space points are necessary to avoid collision of the whole tool body or its envelope. Since safe corners are only sufficient for point safe trajectory, that is, tool tip point, in order to consider the whole tool body envelope some extra points are needed for collision avoidance. These points are generated randomly in the free space area at some random distances as shown in Figure 13.

Each and every image/frame at its time instant is converted to a binary matrix and is updated to provide new corners and safe points as explained in the above two steps.

- *Safe trajectory generation:* The safe trajectory generation step generates safe trajectories when a collision is detected as shown in the illustration example in Figure 13. When collision is detected at the next

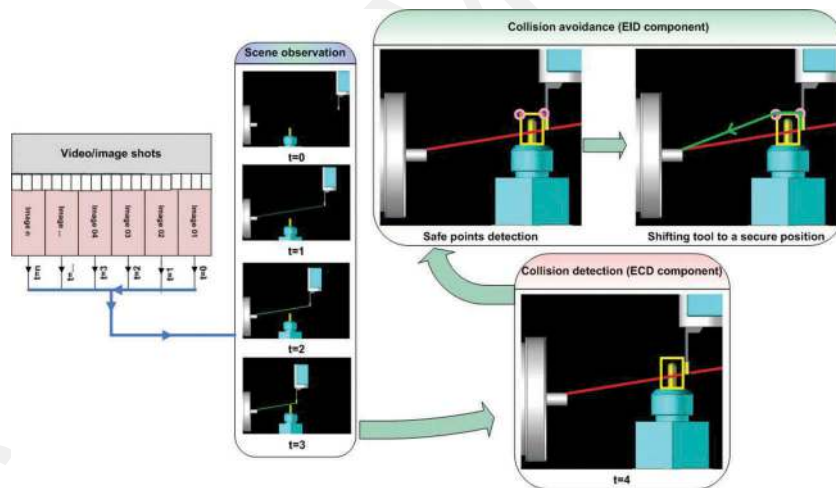


Figure 11. EIDC functionality for a virtual multi-axis machine scenario.

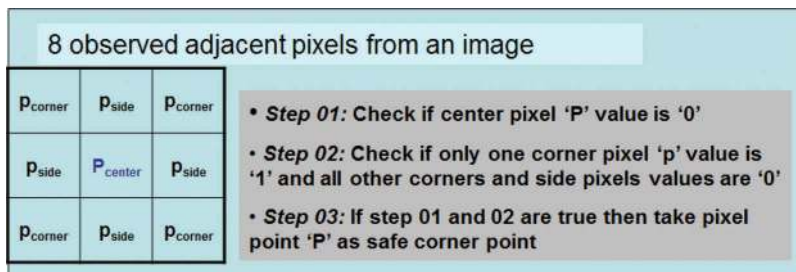


Figure 12. Safe corners detection.

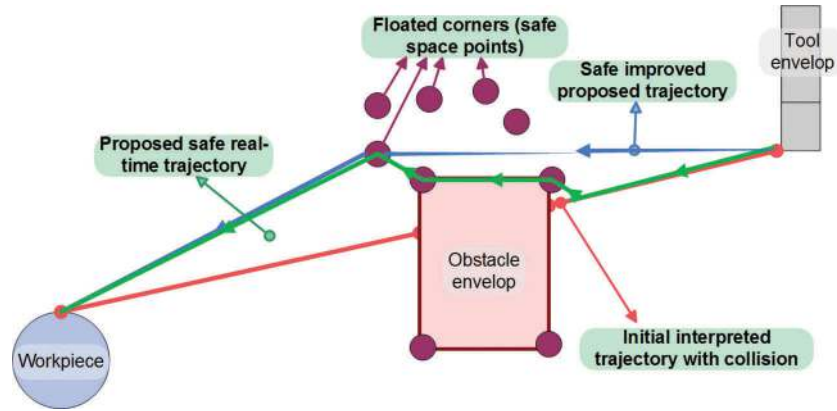


Figure 13. Safe corners and space points detection.

trajectory point, the safe trajectory generation step shifts the tool to a nearest safe point (corner/space) by satisfying certain criteria of safe point selection as formulated below:

- The selected safe point must give the minimum 'addition distance'

addition distance = distance from current tool trajectory point to safe point + distance from safe point to the final destination goal point, as shown in the schematic diagram in Figure 14.

- There is no collision between the current point and the selected safe point.

Results and discussions

Notion of acceleration and deceleration

A scenario has been created, as introduced in the previous section, in order to validate the robustness of the vision based system presented here for rapid collision detection

and avoidance in a virtual environment. An Index G300 virtual multi-axis machine model is considered as an example for the vision system application as shown in Figure 15. A production tool (Tool 01) with a non-functional trajectory is moving towards its point of production with a velocity of 24 mm/s, while another tool (Tool 02, obstacle) is moving into its trajectory plan with 10 mm/s. When both trajectories are verified in virtual CAM software, no collision was detected and both tools passed nearby safely. However, one of the real machines' characteristics (i.e. acceleration/ deceleration) is integrated into the scenario collision is encountered as shown in Figure 16. This collision occurs because of the integration of real characteristics, that is, acceleration and deceleration of real machine-tool (real scene) that are not taken into account by current CAM software (virtual scene). The acceleration and deceleration value is taken as $\gamma = 5000 \text{ mm/s}^2$ according to the machine characteristics (operator experience). The normal time taken by the tool to travel from initial point to the final point without acceleration was 9.6 s for a 230.46 mm length of

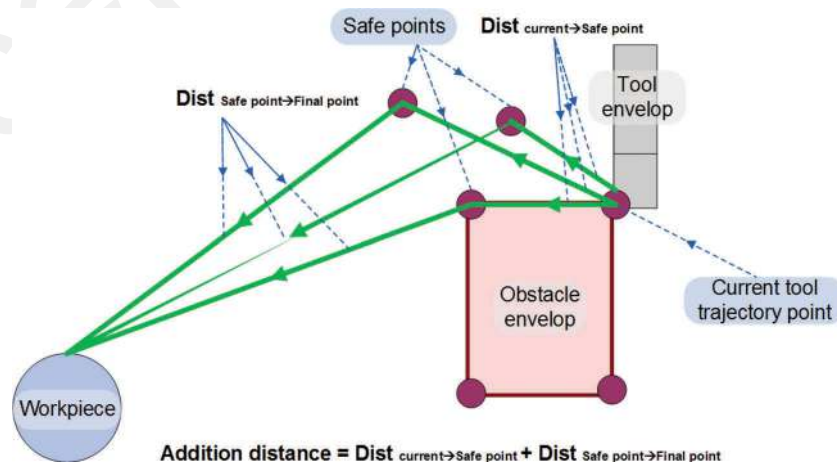


Figure 14. Schematic-diagram of Addition distance.

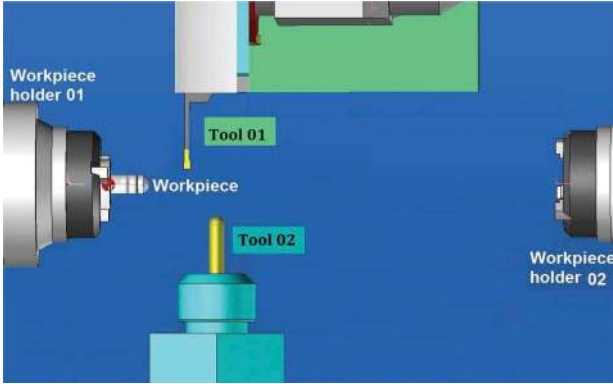


Figure 15. Test virtual machine (Index G300 Turn-Mill machine virtual model).

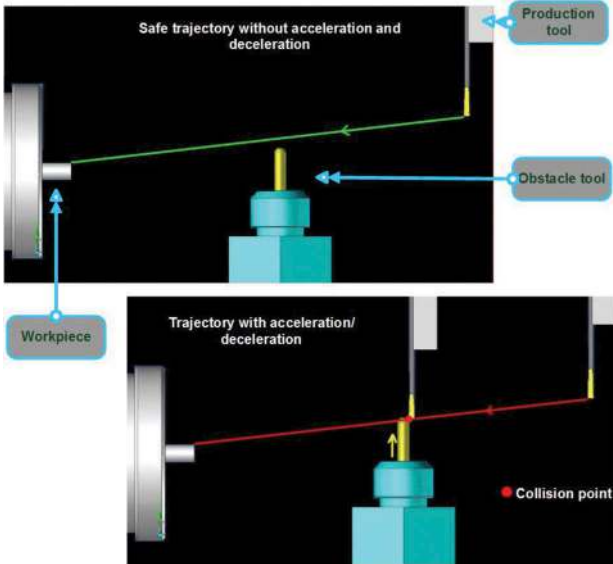


Figure 16. Tool trajectory with or without acceleration/ deceleration.

trajectory. But after acceleration and deceleration integration, the time mounts to 10.35 s. This time increase during tool movement caused collision because meanwhile the other obstacle tool arrives in the trajectory plan. The effect of acceleration and deceleration for theoretical and real-time delay in time is shown in the graphs in Figure 17.

Our approach proposes two solutions to the collision problem discussed in the previous scenario.

- Either increase/decrease the tool speed in order to pass the obstacle without arriving at the point of collision at the same time.

Avoid tool collision by guiding a tool to a safe point. Increasing the tool speed gives a safe trajectory like the

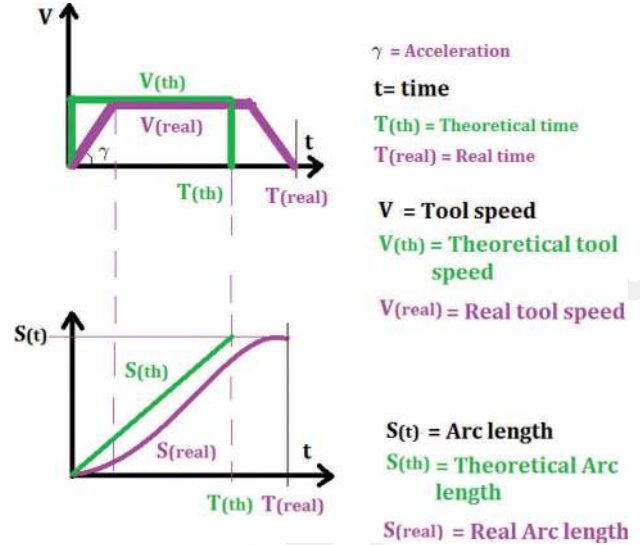


Figure 17. Notion of acceleration/deceleration in virtual (theoretical) and real scene.

first image in Figure 16 and is therefore not shown here but the second solution is treated in more detail.

Safe distance to the obstacle is calculated by the formula given in the previous section and therefore for the same example with tool speed of $\vec{v} = 750 \text{ mm/s}$ (maximum speed) and acceleration of $\gamma = 5000 \text{ mm/s}^2$, the safe distance required to the obstacle should be $\text{Dist}_{\text{safe}} = 56.25 \text{ mm}$ or 239.51 pixels. Similarly, for a tool speed of $\vec{v} = 333.33 \text{ mm/s}$ and acceleration of $\gamma = 5000 \text{ mm/s}^2$, the safe distance required to the obstacle should be $\text{Dist}_{\text{safe}} = 11.11 \text{ mm}$ or 47.31 pixels, whereas, for a tool speed of $\vec{v} = 100 \text{ mm/s}$ and acceleration of $\gamma = 5000 \text{ mm/s}^2$, the safe distance required to the obstacle should be $\text{Dist}_{\text{safe}} = 1 \text{ mm}$ or 4.25 pixels. Therefore, the distance in-between obstacle and its envelope should be considered according to the safe distance required which is a function of relative speeds of tool and obstacle.

Experiment

An experiment was conducted using virtual environment in order to validate the real-time safe trajectory generation system presented in this paper for multi-axis dynamic scenarios. The parameters and results of the experiment are:

- Image size = 1222×740 pixels
- Number of frames considered and the treatment time taken = 11 frames per 5 s for tool tip point safe trajectory generation and 10 frames per 6 s for tool envelope safe trajectory generation
- Tool speed considered $\rightarrow \vec{v} = 100 \text{ mm/s}$.

Let us suppose that the obstacle tool will move with constant speed for simplicity and that the production tool changes its position to a new secure position in the case of collisions. All data regarding the scenario is obtained from different counterparts of the manufacturing system such as CAD, CAM or G-code, whereas object detection and identification along with their speed data is obtained from the vision system.

The solution obtained for the current collision problem scenario by applying the INSTATE algorithms based on the INSTATE system, generated in Matlab (computing software) is shown in Figure 18(a) and (b). The breakdown of result shown in Figure 18(a), is shown in Figure 19. Figure 18(a) shows safe trajectory generated for tool tip point, whereas, Figure 18(b) gives tool envelope trajectory. In both results shown, the obstacle envelope is used for security margin, whereas the tool envelope shown in Figure 18(b) is used for generation of actual tool trajectory.

For the tool envelope trajectory, all four corner point trajectories are verified according to the same procedure for point trajectory and in the case of collision a safe trajectory point is achieved. The small white circle points shown on the trajectory indicate the time of introduction of fresh images into the system for treatment.

As shown in Figures 18(a) and (b) and 19, the obstacle tool is in continuous movement within each time interval

(for changing images, e.g., between $t = 0$ and $t = 1$), but the obstacle envelope is considered as static during each image interval of time. When there is time to change an image after each interval, the obstacle envelope jumps to the new position where the tool will finally arrive within the next time interval. Therefore, if the time interval is larger, the tool envelope should be greater also. Since the size of the envelope is decided by a security margin, it is therefore directly related to the speed of the obstacle, the acceleration of machine and the time interval for each image change. The obstacle envelope makes the process easier and faster by providing safe envelope corners in the space outside the security margin which are used for safe point selection when collision is detected with an obstacle envelope.

The real contribution of this paper is an automatic collision detection and avoidance system implemented in a dynamic virtual simulation environment where conventional systems can detect collisions but are unable to avoid these automatically. To the authors' knowledge, no other such system exists in the literature which automatically decides a safe path for virtual or real multi-axis production machines in a dynamic machining environment. The system not only uses information from virtual scenes but also real machine data to get the actual information of the machine in order to take an intelligent and automatic decision regarding safe tool-paths in virtual environments.

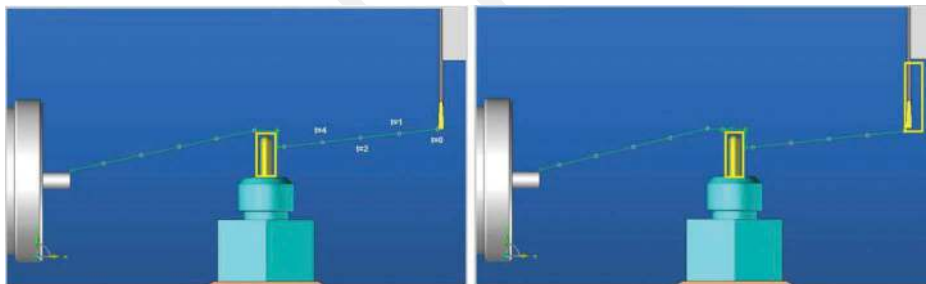


Figure 18. INSTATE results – 1222 × 740 pixels each image: (a: left) with obstacle envelope and point trajectory – 11 frames in 5 s; (b: right) with obstacle envelope and tool envelope trajectory – 10 frames in 6 s.

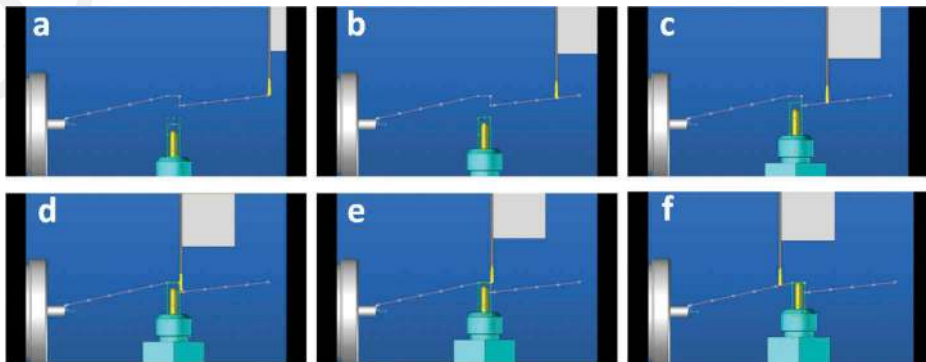


Figure 19. Breakdown of trajectory results shown in Figure 18(a).

The system thus works autonomously without any human intervention.

Perspectives to multi-tool trajectories generation

The results shown in the results section are based on programming one tool at a time. The aspect of multi-tool and multi-channel synchronisation and the related collision challenges will be treated in future work. For instance, multiple robots working and synchronisation (Warren 1990) might be a possible course to follow where robots are actually prioritised (Buckley and May 1989) in order to avoid collisions.

This paper focuses on vision based collision detection and avoidance in a virtual production dynamic environment. The concept presented is therefore applied to a conventional virtual system initially. The main application of this concept will be in the step-compliant data interface for numeric controls (STEP-NC) (Hascoet, Rauch, and Suh' 2007; Rauch et al. 2012), based multi-axis machining environment because both STEP-NC and the proposed vision based concept agree in terms of information accessibility and sharing during manufacturing processes. Hence, a significant need for technology change has been identified where it would be easier to capture information for scenarios and, most importantly, the two way communication between virtual and real machining. The perspective application of the INSTATE system into the STEP-NC environment will also include the structuring of information, development of rules for object movement in the scene for expert decisions to get a collision-free environment and collision detection and avoidance on real machines in real-time.

Modern industry is orienting towards dry manufacturing and laser assisted machining processes due to high costs of the coolant used in manufacturing. The presence of lubricant, dust and chips also impacts the scene clarity especially in functional trajectories. These aspects of manufacturing will be treated in future.

Conclusions

The research work presented here is based on the application of vision based concept for safe and intelligent trajectory generation in dynamic machining environments. The contribution of this paper is to take into account multi-axis machine dynamic characteristics and constraints for an expert decision system to detect and avoid collisions in 2D virtual simulation automatically using a vision based approach where currently no solution is available. An INSTATE production system is developed for safe tool-path generation and correction in a multi-tool machining environment. This vision concept uses video images from virtual simulation, identifies collision scenario, detects and identifies objects, collects information

regarding objects speed and localisation from machine and image processing tools. The proposed approach then uses all this information for generating safe future trajectories by avoiding potential collision risks. Validation results from the vision system focus on a virtual 2D dynamic environment with non-functional, transversal trajectories initially. Future work will focus on applying this system concept into a real machining environment where online collisions will be detected and avoided using sophisticated vision based equipment for more complex scenarios and integrated into STEP-NC based controllers.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

The authors would like to thank Centre National de la Recherche Scientifique (CNRS), France for funding the project.

References

- Ahmad, R., S. Tichadou, and J. Y. Hascoet. 2010. "Integration of Vision Based Image Processing for Multi-Axis CNC Machine Tool Safe and Efficient Trajectory Generation and Collision Avoidance." *Journal of Machine Engineering* 10 (4): 53–65.
- Ahmad, R., S. Tichadou, and J.-Y. Hascoet. 2012. "New Computer Vision Based Snakes and Ladders Algorithm for the Safe Trajectory of Two Axis CNC Machines." *Journal of Computer Aided Design* 44 (5): 355–366. doi:10.1016/j.cad.2011.12.008.
- Ahmad, R., S. Tichadou, and J.-Y. Hascoet. 2013. "3D Safe and Intelligent Trajectory Generation for Multi-Axis Machine Tools Using Machine Vision." *International Journal of Computer-Integrated Manufacturing* 26 (4): 365–385. doi:10.1080/0951192X.2012.717720.
- Ata, A., and T. Myo. 2006. "Collision-Free Trajectory Planning for Manipulators Using Generalized Pattern Search." *International Journal of Simulation and Modelling* 5 (4): 145–154. doi:10.2507/IJSIMM05(4)2.074.
- Balasubramaniam, M., S. Ho, S. Sarma, and Y. Adachi. 2002. "Generation of Collision-Free 5-Axis Tool Paths Using a Haptic Surface." *Computer-Aided Design* 34 (4): 267–279. doi:10.1016/S0010-4485(01)00057-4.
- Bennewitz, M., and W. Burgard. 2000. "A Probabilistic Method for Planning Collision-Free Trajectories of Multiple Mobile Robots." In *Proceedings of the Workshop Service Robotics – Applications and Safety Issues in an Emerging Market at the 14th ECAI*, Vol. 9. <http://ais.informatik.uni-freiburg.de/publications/papers/bennewitz00ecaiws.pdf>
- Bohez, E. L. J., N. T. H. Minh, B. Kiatsrithanakorn, P. Natasukon, H. Ruei-Yun, and L. T. Son. 2003. "The Stencil Buffer Sweep Plane Algorithm for 5-Axis CNC Tool-Path Verification." *Computer-Aided Design* 35 (12): 1129–1142. doi:10.1016/S0010-4485(02)00209-9.
- Buckley, S., and J. May. 1989. "Fast Motion Planning for Multiple Moving Robots." *Robotics and Automation Proceedings, 1989 IEEE International Conference* 1: 322–326.

- Castro, J. L., M. Delgado, J. Medina, and M. D. Ruiz-Lozano. 2011. "An Expert Fuzzy System for Predicting Object Collisions. Its Application for Avoiding Pedestrian Accidents." *Expert Systems with Applications* 38 (1): 486–494. doi:10.1016/j.eswa.2010.06.088.
- Dugas, A., J. J. Lee, M. Terrier, and J. Y. Hascoet. 2002. "Virtual Manufacturing for High Speed Milling." 35th CIRP International Seminar on Manufacturing Systems, Séoul, May 13–15, 199–206.
- Hascoet, J. Y., F. Bennis, and F. Guerin. 1993. "Optimization of Tool Trajectory for Five Axis Control Machining." International Conference on IASTED Robotics and Manufacturing, Oxford, September 23–25, 120–122. ISBN 0 88986 179 X.
- Hascoet, J. Y., M. Rauch, and S. Suh. 2007. "Relevance of STEP-NC Standard for High Level Toolpaths Generation." International Conference on High Speed Milling, San Sebastian, March 21–22.
- Heidelberger, B., M. Teschner, and M. Gross. 2004. "Detection of Collisions and Self-Collisions Using Image-Space Techniques." *Journal of WSCG* 12 (3): 145–152.
- Ho, S., S. Sarma, and Y. Adachi. 2001. "Real-Time Interference Analysis between a Tool and an Environment." *Computer-Aided Design* 33 (13): 935–947. doi:10.1016/S0010-4485(00)00117-2.
- Jiménez, P., F. Thomas, and C. Torras. 2001. "3D Collision Detection: A Survey." *Computers & Graphics* 25 (2): 269–285. doi:10.1016/S0097-8493(00)00130-8.
- Jun, C.-S., K. Cha, and Y.-S. Lee. 2003. "Optimizing Tool Orientations for 5-Axis Machining by Configuration-Space Search Method." *Computer-Aided Design* 35 (6): 549–566. doi:10.1016/S0010-4485(02)00077-5.
- Lee, S., and G. Kardaras. 1997. "Elastic String Based Global Path Planning Using Neural Networks." In *Proceedings of IEEE International Symposium on Computational Intelligence in Robotics and Automation, CIRA'97*, 108–114. IEEE. doi:10.1109/CIRA.1997.613846.
- Makhanov, S. 2007a. "Optimization and Correction of the Tool Path of the Five-Axis Milling Machine: Part 1. Spatial Optimization." *Mathematics and Computers in Simulation* 75 (5–6): 210–230. Applied Scientific Computing: Advanced Grid Generation. Approximation and Simulation. doi:10.1016/j.matcom.2006.12.009
- Makhanov, S. 2007b. "Optimization and Correction of the Tool Path of the Five-Axis Milling Machine: Part 2: Rotations and Setup." *Mathematics and Computers in Simulation* 75 (5–6): 231–250. Applied Scientific Computing: Advanced Grid Generation, Approximation and Simulation. doi:10.1016/j.matcom.2006.12.001
- Moravec, H., and A. Elfes. 1985. "High Resolution Maps from Wide Angle Sonar." In *Proceedings of 1985 IEEE International Conference on Robotics and Automation*, Vol. 2, 116–121. IEEE. doi:10.1109/ROBOT.1985.1087316.
- Morishige, K., K. Kase, and Y. Takeuchi. 1997. "Collision-Free Tool Path Generation using 2-Dimensional C-Space for 5-Axis Control Machining." *The International Journal of Advanced Manufacturing Technology* 13: 393–400. doi:10.1007/BF01179033.
- Moriwaki, T. 2008. "Multi-Functional Machine Tool." *CIRP Annals – Manufacturing Technology* 57 (2): 736–749. doi:10.1016/j.cirp.2008.09.004.
- Petiot, J. F., P. Chedmail, and J. Y. Hascoet. 1998. "Contribution to the Scheduling of Trajectories in Robotics." *Robotics and Computer-Integrated Manufacturing* 14: 237–251. doi:10.1016/S0736-5845(97)00032-X.
- Rauch, M., R. Laguionie, J.-Y. Hascoet, and S.-H. Suh. 2012. "An Advanced STEP-NC Controller for Intelligent Machining Processes." *Robotics and Computer-Integrated Manufacturing* 28: 375–384. doi:10.1016/j.rcim.2011.11.001.
- Sasiadek, J. Z., and I. Duleba. 2000. "3D Local Trajectory Planner for UAV." *Journal of Intelligent and Robotic Systems* 29: 191–210. doi:10.1023/A:1008108910932.
- Tang, T., E. L. Bohez, and P. Koomsap. 2007. "The Sweep Plane Algorithm for Global Collision Detection with Workpiece Geometry Update for Five-Axis NC Machining." *Computer-Aided Design* 39 (11): 1012–1024. doi:10.1016/j.cad.2007.06.004.
- Wang, Q.-H., J.-R. Li, and -R.-R. Zhou. 2006. "Graphics Assisted Approach to Rapid Collision Detection for Multi-Axis Machining." *The International Journal of Advanced Manufacturing Technology* 30: 853–863. doi:10.1007/s00170-005-0127-5.
- Warren, C. W. 1990. "Multiple Robot Path Coordination Using Artificial Potential Fields." In *Proceedings of IEEE International Conference on Robotics and Automation*, Vol. 1, 500–505. IEEE. doi:10.1109/ROBOT.1990.126028.
- Zhang, L., X. Huang, Y. J. Kim, and D. Manocha. 2008. "D-Plan: Efficient Collision-Free Path Computation for Part Removal and Disassembly." *Journal of Computer-Aided Design and Applications* 5: 1–4. doi:10.3722/cadaps.2008.xxx-yyy.