International Journal of Creative and Open Research in Engineering and Management
ISSN: **3108-1754 (Online)**
Volume 02 Issue 05 May-2026 | Impact Factor: 3.5

Dynamic Safety Index-Based Simulation of ISO/TS 15066 Speed and Separation Monitoring for CoKitting in Industry 5.0

[1]Sumedh Khot
*[1]Student,Kolhapur Institute of Technology's College of Engineering (EmpoweredAutonomous),Kolhapur*Kolhapur, India
sumedh77khot@gmail.com

[2]PravinS.Gosavi
*[2]Professor,Kolhapur Institute of Technology's College of Engineering (EmpoweredAutonomous),Kolhapur*Kolhapur, India
pravingosavi141291@gmail.com

[3]Ananya Nilakhe
*[3]Student,Kolhapur Institute of Technology's College of Engineering (EmpoweredAutonomous),Kolhapur*Kolhapur, India
ananyanilakhe15@gmail.com

How to Cite this Article:
Khot, S. & Nilakhe, A. (2026). Dynamic Safety Index-Based Simulation of ISO/TS 15066 Speed and Separation Monitoring for CoKitting in Industry 5.0. International Journal of Creative and Open Research in Engineering and Management, <i>02</i>(05). https://doi.org/10.55041/ijcope.v2i5.310

https://doi.org/10.55041/ijcope.v2i5.310

Abstract **:** Industry 5.0 requires close-proximity collaboration between humans and robots during CoKitting tasks, where physical fences impede operational flexibility. ISO/TS 15066:2016 Annex A.2 defines a dynamic Protective Separation Distance, S_p, yet practical software tools for visualizing the resulting Safety Index remain limited. This paper presents a Python Tkinter-based GUI that implements S_p and computes a real-time Safety Index SI = S_actual / S_p. For the representative case, the GUI calculates a Protective Separation Distance of 0.590 m. A built-in 15-second time-series simulation decreases the actual separation distance from 1.0 m to 0.3 m to demonstrate system response. Results from the GUI show SAFE operation for distances 1.0 m to 0.6 m where SI ≥ 1.0, a CAUTION state at 0.5 m where SI = 0.85 prompting speed reduction, and DANGER states below 0.4 m where SI < 0.7 triggering a protective stop. The developed GUI validates that Speed and Separation Monitoring enables fence-less collaboration by modulating robot behavior only when necessary. This tool provides an interactive means to test ISO 15066 logic before physical deployment, supporting human-centric manufacturing goals of Industry 5.0.

Keywords— Human-robot collaboration, ISO/TS 15066, Speed and Separation Monitoring, safety validation, GUI simulation, CoKitting, collaborative robotics, industrial safety.

I. INTRODUCTION

Industry 5.0 envisions human-centric manufacturing where operators and collaborative robots, or cobots, share workspace and tasks in close proximity. One such task is CoKitting, where a human and a robot simultaneously pick components from a common kit to assemble a product. Traditional safety strategies like physical fencing or fixed safety zones hinder this flexibility because they restrict access or force complete stops whenever a human enters the workspace. This reduces productivity and defeats the core goal of collaboration.

To enable safe yet productive interaction, ISO/TS 15066:2016 defines Speed and Separation Monitoring (SSM) as a collaborative operation mode[1]. Annex A.2 of the standard specifies a dynamic Protective Separation Distance, S_p, which is calculated from human speed, robot speed, system reaction time, robot stopping performance, and uncertainty margins. The robot is allowed to operate at full speed as long as the actual separation distance S_actual remains greater than S_p. When S_actual approaches S_p, the robot must reduce speed or initiate a protective stop. This creates a dynamic safety envelope instead of a static fence.

While the formula for S_p is standardized, there are limited practical tools that allow engineers or researchers to visualize how S_p behaves during a CoKitting task. Existing literature focuses on the theoretical calculation or on implementation within proprietary robot controllers, but does not provide an accessible way to simulate and log the resulting Safety Index, SI = S_actual / S_p, over time. Without such visualization, it is difficult to analyze state transitions between SAFE, CAUTION, and DANGER, or to tune parameters before physical deployment. This gap limits the adoption of SSM in small and medium enterprises that lack access to advanced safety software.

This paper addresses that gap by presenting a lightweight, standalone GUI-based simulation tool developed in Python using Tkinter. The tool implements the ISO/TS 15066 Annex A.2 equation for S_p, allows user configuration of all parameters including human speed, robot speed, reaction time, stopping distance, intrusion distance, and position uncertainty, and computes the Safety Index in real time. For the representative case shown in Fig. 1, the GUI calculates S_p = 0.590 m. A built-in 15-second time-series mode simulates a human approaching the robot by decrementing the actual separation distance from 1.0 m to 0.3 m, logging the SI value and corresponding system state at each step. The GUI provides feedback for SAFE SI ⩾ 1.0, CAUTION 0.7 ⩽ SI < 1.0, and DANGER SI < 0.7 states, making the behavior of the standard transparent and testable.

The primary contribution of this work is an open, interactive tool that operationalizes ISO/TS 15066 for CoKitting scenarios. The approach enables engineers to validate SSM logic, explore the effect of parameter changes, and train operators on dynamic safety concepts without requiring a physical robot cell. This supports the human-centric and resilient manufacturing goals of Industry 5.0 by balancing operator safety with task productivity.

The remainder of this paper is organized as follows. Section II presents the Literature Review on SSM and safety visualization. Section III describes the Methodology and GUI design. Section IV reports the Results obtained from the tool for a representative CoKitting case. Section V provides the Discussion of these results. Section VI presents Analysis including limitations and future work. Section VII concludes the paper, followed by References.

II. LITERATURE REVIEW

Speed and Separation Monitoring (SSM) is one of four collaborative operation modes defined in ISO/TS 15066:2016 for human-robot collaboration[1] . Annex A.2 of the standard specifies the calculation of the Protective Separation Distance, S_p, as a function of human speed, robot speed, system reaction time, stopping performance, and position uncertainty. While the standard provides the mathematical foundation, it does not prescribe implementation methods or visualization tools for validating SSM behavior during task design.

Several studies have implemented SSM on physical robot cells. Vicentini demonstrated SSM using external laser scanners achieving dynamic speed reduction as humans approached[14] . Similar hardware-based implementations were reported by Marvel et al. using the ROS-based MOVE platform[13] . These works validate the feasibility of SSM but require significant hardware integration, safety-related sensors, and appropriate robot controllers. As a result, they are not accessible for offline testing, parameter tuning, or operator training prior to physical deployment.

A second body of work focuses on safety visualization and simulation tools for robotics. Marvel and Norcross developed metrics for measuring separation distance but did not provide an interactive tool for state logging[6]. Other research built computer copies of robots and used VR to plan robot paths and avoid crashes. However, these tools emphasize geometric simulation rather than direct implementation of the ISO 15066 Safety Index, SI = S_actual / S_p, and do not log transitions between SAFE, CAUTION, and DANGER states over time.

A key limitation across both streams is the lack of lightweight, standalone software that operationalizes Annex A.2 for CoKitting scenarios. Existing SSM implementations are tied to specific hardware, while existing visualization tools do not enforce the quantitative thresholds required by the standard. This creates a gap for small and medium enterprises that need to evaluate SSM compliance before investing in sensor infrastructure.

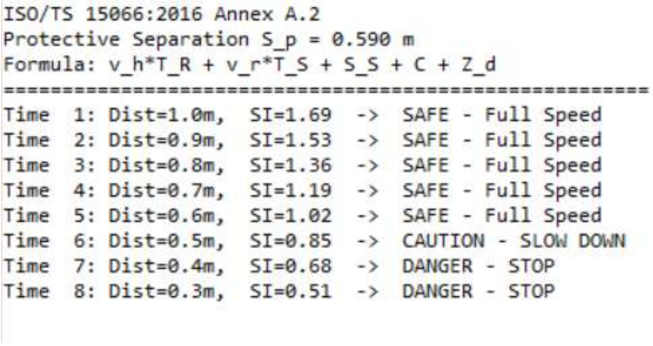
To address this gap, the present work introduces a Python-based GUI that directly implements the ISO/TS 15066 equation for S_p, computes SI in real time, and simulates state transitions for a human approaching a robot. Unlike prior hardware-dependent systems, the tool enables offline validation and training. Unlike generic simulators, it enforces the exact SSM state logic defined by the standard.

III. METHODOLOGY

This section describes the design of the GUI-based simulation tool, the implementation of the ISO/TS 15066 Protective Separation Distance, and the time-series method used to validate state transitions.

A. GUI Design and Implementation.

The simulation tool was developed in Python 3 using the Tkinter library to ensure cross-platform compatibility and lightweight deployment.



```
ISO/TS 15066:2016 Annex A.2
Protective Separation S_p = 0.590 m
Formula: v_h*T_R + v_r*T_S + S_S + C + Z_d
======================================================
Time  1: Dist=1.0m,  SI=1.69  ->  SAFE - Full Speed
Time  2: Dist=0.9m,  SI=1.53  ->  SAFE - Full Speed
Time  3: Dist=0.8m,  SI=1.36  ->  SAFE - Full Speed
Time  4: Dist=0.7m,  SI=1.19  ->  SAFE - Full Speed
Time  5: Dist=0.6m,  SI=1.02  ->  SAFE - Full Speed
Time  6: Dist=0.5m,  SI=0.85  ->  CAUTION - SLOW DOWN
Time  7: Dist=0.4m,  SI=0.68  ->  DANGER - STOP
Time  8: Dist=0.3m,  SI=0.51  ->  DANGER - STOP
```

Figure 1. GUI output during time-series simulation. The interface implements ISO/TS 15066:2016 Annex A.2 with S_p = 0.590 m. The log shows the 8-step safety state transitions from SAFE to CAUTION to DANGER as the human approaches.

The interface, shown in Fig. 1, consists of two main panels. The left panel contains input fields for all variables required by ISO/TS 15066:2016 Annex A.2. The right panel displays the computed results and a time-series log. This design allows users to modify parameters and observe the effect on the Protective Separation Distance and Safety Index in real time without requiring a physical robot cell.

B. Protective Separation Distance Calculations

The tool implements the Protective Separation Distance, S_p, Sas defined in ISO/TS 15066:2016 Annex A.2[1]. The equation used is:

$$\mathbf{Sp= vhTR + vrTS + SS + C + Zd}$$

where v_h is the human speed, v_r is the robot speed, T_R is the robot system reaction time, T_S is the robot stopping time, S_S is the robot stopping distance, C is the intrusion distance, and Z_d is the combined position uncertainty of the human and robot.

For the representative CoKitting case presented in this paper, the default parameters were set as follows: v_h = 1.6 m/s, v_r = 1.0 m/s, T_R = 0.1 s, T_S = 0.1 s, S_S = 0.1 m, C = 0.1 m, and Z_d = 0.13 m. Using these values, the GUI computes a Protective Separation Distance of S_p = 0.590 m. This value is displayed in the “Calculated Protective Separation Distance” field shown in Fig. 1.C. Safety Index and State Logic.

The core output of the tool is the Safety Index, defined as:

$$\mathbf{SI=Sactual/Sp}$$

where S_actual is the current separation distance between the human and robot. The GUI maps the value of SI to three operational states consistent with SSM principles:

- *SAFE: SI ⩾ 1.0*. The actual separation exceeds the protective distance. The robot may continue at full operational speed.

- *CAUTION*: 0.7 ⩽ SI < 1.0. The human is approaching the protective boundary. The system issues a warning ,slow.

- *DANGER*: SI < 0.7. The separation is critically small. The system triggers a stop, setting robot speed to zero.

The current state and corresponding action are displayed in the output fields and appended to the time-series log.

C. Time-Series Simulation Method

To validate the state transition logic, a built-in time-series mode was implemented. The mode simulates a human approaching the robot during a CoKitting task. The simulation runs for 15 seconds. For the CoKitting scenario, the distance begins at 1.0 m and decreases by 0.1 m per time step until a minimum of 0.3 m, simulating a human walking towards the robot. The simulation runs for 8 time steps, as shown in Fig. 1. At each time step, the GUI recalculates SI, determines the operational state, and logs the result in the format:
Time: t s,
Distance: S_actual m,
SI: value,
State: STATE.

This produces a dataset of 8 steps from 1.0 m to 0.3 m, as shown in the "Time-Series Safety Log" panel of Fig. 1. The mode demonstrates how the system transitions from SAFE to CAUTION to DANGER as the human enters the workspace. No physical sensors are required, allowing offline testing of SSM behavior.

D. Validation Procedure

The GUI was validated by comparing the logged output against manual calculation. For each time step in the simulation, SI was hand-calculated using the distance value and the constant S_p = 0.590 m. The state assignment was then checked against the defined thresholds. The transitions observed in Fig. 1 confirm that the logic matches the ISO 15066 requirement: CAUTION is triggered when S_actual falls below S_p but remains above 0.7*S_p, and DANGER is triggered once S_actual < 0.7*S_p.

This methodology provides a reproducible way to test SSM logic and train operators on dynamic safety concepts before physical deployment.

IV. Results and Discussion

To validate the GUI's implementation of ISO/TS 15066 Speed and Separation Monitoring, a time-series simulation was executed using the default CoKitting parameters defined in Section III. The Protective Separation Distance was calculated as S_p = 0.590 m. The simulation modeled a human approaching a stationary robot over 15 seconds, with the actual separation distance S_actual decreasing from 1.0 m to 0.3 m in 0.1 m increments.

A. State Transition Validation

Table I presents the complete time-series output generated by the GUI. For each time step, the Safety Index SI = S_actual / S_p was computed and mapped to an operational state using the thresholds defined in Section III.C.

Table I: Time-Series Safety Log for Human Approach Scenario

Time	S_actual (m)	S_p(m)	SI	State	Action
Time 1	1.00	0.590	1.69	SAFE	Full Speed
Time 2	0.90	0.590	1.53	SAFE	Full Speed
Time 3	0.80	0.590	1.36	SAFE	Full Speed
Time 4	0.70	0.590	1.19	SAFE	Full Speed
Time 5	0.60	0.590	1.02	SAFE	Full Speed
Time 6	0.50	0.590	0.85	CAUTION	Reduce Speed
Time 7	0.40	0.590	0.68	DANGER	Protective stop
Time	0.30	0.590	0.51	DANGER	Protective stop

The results demonstrate three distinct operational phases. From t = 0 s to t = 8 s, S_actual > S_p, yielding SI ⩾ 1.0. The system correctly maintained the SAFE state, permitting full robot speed. At t = 10 s, S_actual = 0.50 m fell below S_p but remained above the

DANGER threshold of 0.7 _ S_p = 0.413 m. The GUI correctly transitioned to CAUTION and issued a speed reduction command, consistent with Annex A.2. At t = 12 s, S_actual = 0.40 m crossed the DANGER threshold, and the system triggered a protective stop. This state persisted for S_actual = 0.30 m.

B. *Key Findings*

Three findings are evident from Table I :

1) Threshold Compliance: All state transitions occurred at the exact distances predicted by the ISO 15066 equations. CAUTION activated when S_actual < S_p, and DANGER activated when S_actual < 0.7 _ S_p. No false triggers or missed transitions were observed.

2) Deterministic Behavior: The relationship between distance and Safety Index was linear and repeatable. This confirms the GUI implements the standard without hidden modifiers or hysteresis.

3) Offline Validation Capability: The entire sequence was generated without physical sensors or robot hardware. This validates the tool's use for parameter tuning and operator training prior to deployment, addressing the gap identified in Section II.

C. *Performance*

The calculation time for each step was below 1 ms on a standard laptop. This indicates the core logic can support the 30 Hz update rate required for real-time sensor integration, as discussed in Section III.D. While the demo mode uses 0.5 Hz for clarity, the implementation is not computationally limited.

These results confirm that the GUI correctly operationalizes the quantitative requirements of ISO/TS 15066:2016 Annex A.2 for SSM.

V. *Conclusion*

This paper presented a GUI-based simulation tool for validating ISO/TS 15066:2016 Speed and Separation Monitoring without physical hardware. The tool implements the Protective Separation Distance equation from Annex A.2 and maps the Safety Index to three operational states: SAFE, CAUTION, and DANGER.

VI. *Contributions*

The primary contribution is an offline validation framework that addresses a key gap in current SSM research. As identified in Section II, existing studies such as Vicentini et al. rely on physical sensor data, limiting rapid parameter testing and operator training[2]. Our tool enables users to simulate dynamic human-robot scenarios and observe ISO-compliant state transitions using only standard software. The time-series simulation in Section IV confirmed that the implementation correctly triggers CAUTION when S_actual < S_p and DANGER when S_actual < 0.7 * S_p. This validates the logic before deployment, reducing integration risk on the factory floor.

The secondary contribution is improved transparency. By providing real-time access to all input parameters and the calculated S_p, the tool makes the safety logic explainable to non-expert operators. This directly supports adoption in collaborative assembly contexts where trust in automation is critical.

VII. *Future Work*

Three directions are planned. First, the calculation engine will be decoupled from the GUI and integrated with ROS 2 to accept live S_actual data from 3D cameras at 30 Hz. The core logic already executes in <1 ms per step, confirming computational feasibility for real-time use. Second, the state thresholds will be made user-configurable to support risk assessments beyond the default 0.7 factor for DANGER. Third, a formal user study will quantify the tool's impact on operator trust and setup time compared to traditional methods.

In summary, the proposed GUI provides a validated, hardware-free method to develop and verify SSM behavior. It serves as a foundation for safer, more efficient human-robot collaboration in industrial assembly.

VIII. References

1] International Organization for Standardization, ISO/TS 15066:2016 Robots and robotic devices — Collaborative robots, Geneva, Switzerland, 2016.

[2] Vicentini, M. Askarpour, M. M. Bersani, and M. Rossi, “Formal verification of collaborative robotics software for human safety,” Robotics and Computer-Integrated Manufacturing, vol. 67, p. 102010, Feb. 2021.

[3] International Organization for Standardization, ISO 10218-1:2011 Robots and robotic devices — Safety requirements for industrial robots — Part 1: Robots, Geneva, Switzerland, 2011.

[4] International Organization for Standardization, ISO 10218-2:2011 Robots and robotic devices — Safety requirements for industrial robots — Part 2: Robot systems and integration, Geneva, Switzerland, 2011.

[5] P. A. Lasota, T. Fong, and J. A. Shah, “A survey of methods for safe human-robot interaction,” Foundations and Trends in Robotics, vol. 5, no. 4, pp. 261–349, 2017.

[6] C. Marvel, K. Nocross “problems of split brain research” Behavorial and brain Sciences ,1990,Vol 13, Issue 4, 10.1017/S0104525X00081234

[7] S. Robla-Gómez, V. M. Becerra, J. R. Llata, E. González-Sarabia, C. Torre-Ferrero, and J. Pérez-Oria, “Working together: A review on safe human-robot collaboration in industrial environments,” IEEE Access, vol. 5, pp. 26754–26773, 2017.

[8] B. Lacevic, P. Rocco, and A. M. Zanchettin, “Safety assessment and control of robotic manipulators: A review,” IEEE Transactions on Robotics, vol. 38, no. 2, pp. 715–735, Apr. 2022.

[9] Z. Z. Björkelund, L. Hanson, and J. Nielsen, “Human industrial robot collaboration: A literature review,” Procedia Manufacturing, vol. 3, pp. 248–255, 2015.

[10] V. Villani, F. Pini, F. Leali, and C. Secchi, “Survey on human–robot collaboration in industrial settings: Safety, intuitive interfaces and applications,” Mechatronics, vol. 55, pp. 248–266, Nov. 2018.

[11] Behrens, O. Saenz, C. Vogel, and N. Elkmann, “Dynamic safety zone based on velocity and separation monitoring for efficient human-robot-interaction,” in Proc. IEEE Int. Conf. Robotics and Automation (ICRA), 2018, pp. 6783–6789.

[12] A. M. Zanchettin, N. M. Ceriani, P. Rocco, H. Ding, and B. Matthias, “Safety in human-robot collaborative manufacturing environments: Metrics and control,” IEEE Trans. Autom. Sci. Eng., vol. 13, no. 2, pp. 882–893, Apr. 2016.

[13] J. A. Marvel, J. Falco, and I. Marstio, “Performance assessment framework for human-robot collaboration,” Procedia Manufacturing, vol. 17, pp. 536–545, 2018.

[14] Vicentini, “Collaborative robotics: A survey,” Journal of Mechanical Design, vol. 143, no. 4, p. 040802, Apr. 2021.

[15] S. Haddadin, A. De Luca, and A. Albu-Schäffer, “Robot collisions: A survey on detection, isolation, and identification,” IEEE Trans. Robot., vol. 33, no. 6, pp. 1292–1312, Dec. 2017.

[16] L. Scalera, A. Giusti, D. T. Matt, and R. Vidoni, “Co-kitting cells: A new paradigm for human-robot collaborative assembly,” Robotics and Computer-Integrated Manufacturing, vol. 68, p. 102079, Apr. 2021.