Automatic planning and optimization of a laser radar inspection system

International Journal of Computer Integrated Manufacturing

ISSN: 0951-192X (Print) 1362-3052 (Online) Journal homepage: www.tandfonline.com/journals/tcim20

Automatic planning and optimization of a laser radar inspection system

Domenico Spensieri, Jack Sandberg, Edvin Åblad, Johan Torstensson, Jonas Kressin, Daniel Strand & Richard Lindqvist

To cite this article: Domenico Spensieri, Jack Sandberg, Edvin Åblad, Johan Torstensson, Jonas Kressin, Daniel Strand & Richard Lindqvist (30 Apr 2025): Automatic planning and optimization of a laser radar inspection system, International Journal of Computer Integrated Manufacturing, DOI: 10.1080/0951192X.2025.2493624

To link to this article: https://doi.org/10.1080/0951192X.2025.2493624

Published online: 30 Apr 2025.

INTERNATIONAL JOURNAL OF COMPUTER INTEGRATED MANUFACTURING
https://doi.org/10.1080/0951192X.2025.2493624

Automatic planning and optimization of a laser radar inspection system

Domenico Spensieri[a], Jack Sandberg[a], Edvin Åblad[a], Johan Torstensson[a], Jonas Kressin[a], Daniel Strand[b] and Richard Lindqvist[c,b]

[a]Geometry and Motion Planning Group, Fraunhofer-Chalmers Centre, Gothenburg, Sweden; [b]Saab Aeronautics, Linköping, Sweden; [c]Department of Production Engineering, Manufacturing and Metrology Systems, KTH Royal Institute of Technology, Stockholm, Sweden

ABSTRACT

This work advances the field of inspection process planning by introducing novel models and optimization algorithms for off-line planning in laser radar metrology systems, where re-calibration is needed. The key contribution is a method for the automatic cycle time optimization of inspection programs, accounting for the system to be re-calibrated by measuring some of the tooling spheres. Customized algorithms are derived to solve the problem, together with a novel integer programming model based on a set covering approach for viewpoint coverage planning. The proposed method is implemented within an off-line simulation environment, where its effectiveness and limitations are analyzed through a computational study and validated in industrial scenarios.

ARTICLE HISTORY
Received 24 June 2024
Accepted 31 March 2025

KEYWORDS
Inspection; laser sensor; computer-integrated manufacturing; integer programming; algorithms

1. Introduction

Dimensional management and metrology play a critical role in modern manufacturing, particularly with increasing product customization and stringent quality requirements. Evaluating both functional and geometric quality typically involves inspection processes, but due to time and cost constraints, full inspections are often impractical, necessitating sample-based assessments. The primary trade-off in metrology is balancing:

- high-accuracy monitoring of key features vs.
- minimizing cycle times and costs.

Coordinate Measuring Machines (CMMs) remain a gold standard due to their precision and robustness (Dowling et al. 1997; Mahmud et al. 2011; Yau and Menq 1991; Zhao, Xu, and Xie 2009). However, their slow operation and high cost impose strict feature selection to meet production time constraints. The goal is to maximize both the measurement accuracy and the number of inspected key features within the available time.

To address speed limitations, alternative metrology systems have emerged, such as robot-mounted sensors, which are significantly faster and more cost-effective than CMMs while eliminating mechanical contact. Among these, laser radar-based dimensional inspection systems offer high-speed measurements but introduce challenges in programming and accuracy calibration due to the complexity of sensor positioning.

Additionally, increasing quality assurance demands and tighter cycle time constraints drive the need for faster commissioning and measurement processes. In this context, digitalization offers a transformative solution (Archenti et al. 2024; Azamfirei, Psarommatis, and Lagrosen 2023). The integration of CAD models, combined with advances in digital twins, enables off-line process planning, even off-site. Furthermore, automation in digital workflows helps reduce:

- commissioning time for customized measurement programs;
- time required for executing measurements.

This work presents a novel optimization approach to choose a sequence of sensor configurations subject to a re-calibration constraint. Section 2 positions the study within existing research. Section 3 introduces the mathematical model, while Section 4 details algorithms for feasible and optimal solutions. Computational results in Section 5 evaluate solving

CONTACT Domenico Spensieri domenico.spensieri@fcc.chalmers.se

times across diverse instances. Finally, the conclusions highlight practical applications and future research directions.

2. Related work

In this work, the focus is on minimizing the time needed for an inspection system to measure a given set of key features, while fulfilling measurement accuracy criteria. This goal is very common in the literature. In (Glorieux and Franciosa 2020), for example, the authors minimize viewpoint sampling to cover free-form surfaces, thus shortening cycle-time, while still observing all important primitive to guarantee solution quality. In (Liu et al. 2022), the problem is tackled in the reverse way: a novel coverage path planning method is proposed, with an optimal viewpoint sampling to minimize cycle time, which even incorporates measurement uncertainty of key points. In (Sadaoui et al. 2022), CMM-mounted laser sensor orientations are minimized and combined with automatic path planning to reduce the measurement time and increase the measurement quality.

Similarly to the above cited works, in this article the authors strive towards minimizing cycle time through few laser sensor positions, with the additional complication of having coupled reference systems between different measurements, through the *3-sphere* constraint.

There is a large amount of work regarding coverage path planning for robotics (Galceran and Carreras 2013), in several application areas. Many of them are in the field of scanning and 3D object reconstruction, such as (Fan et al. 2016; Jing et al. 2020; Lauri et al. 2020), and (Pan, Hu, and Wei 2022).

Here, however, the main interest is towards inspection processes and, in general, viewpoint optimization where the sensors can be moved by robots or other mechanisms. An overview of automated inspection planning approaches with comparison to manual planning is given in (Gospodnetić et al. 2022). Traditionally, a lot of research has been done to identify areas (or viewpoints) to entirely cover the workpiece to be inspected (Raffaeli et al. 2013), and (Mahmud et al. 2011). One of the first works in the field is (Sheng et al. 2003), where the minimum number of viewpoints is found, considering visibility, field of view, resolution and focus. Also in (Spensieri et al. 2025), the optimal number of sensor configurations is computed, in order to inspect all given key features. Another work dealing with robotic inspection, with control on measurement uncertainty, is (Liu et al. 2022), where both viewpoint sampling and the sequence of the resulting samples are optimized.

A path planning method based on adjacency graphs and Voronoi diagrams is used in (Li et al. 2018) for the inspection of aerospace structures. Minimizing the number of viewpoints based on targeted areas, rather than uniform or random sampling, is proposed in (Glorieux and Franciosa 2020). Their work includes also constraints relative to the robot kinematics and collision avoidance. Inspection tasks where the sensor is carried by robots are investigated also in (Bogaerts et al. 2018), where the path length for robots motions is minimized, while still maintaining the observability of measurement points. Lately, there has also been a focus on using algorithms to provide computer-aided inspection planning (Sadaoui, Mehdi-Souzani, and Lartigue 2018), similarly to our work.

Motion planning in robot inspection has been studied in (Bircher et al. 2015; Jose and Kumar Pratihar 2016). A review of robotic infrastructure for inspection system is given in (Lattanzi and Miller 2017).

Here, a set of features is assumed to be already defined by the inspection engineer. The goal is to identify the minimum number of configurations for the mechanism to position the sensor in a way that all the features are covered. In this article, integer programming techniques are used to model the problem, which are also used in motion planning (Lattanzi and Miller 2017).

On the other hand, in our application, the process imposes that, any time the sensor changes its position, three previously seen tooling spheres need to be seen again, in order to re-calibrate the relative position between the sensor and the inspected product. At the best of the authors' knowledge, there is no work modeling the *3-sphere* constraint in a mathematical optimization approach. The only found similar constraint is the image-based registration, which brings images into a common reference frame (Scott, Roth, and Rivest 2001).,

Similar constraints also arise in CMM applications for inspection (Yau and Menq 1991), and in (Salman et al. 2016) where Traveling Salesman Problem (TSP) extensions have been proposed.

3. Modeling

In this section, the system is described, together with the criteria that need to be satisfied to have a valid measurement of a given feature on the workpiece. Moreover, the mathematical model is introduced by a discretization of the two degrees of freedom (DOFs) relative to the turntable and the vertical axis, together with the definition of some critical sets that will be used in the rest of the paper.

3.1. System description

The investigated mechanical system comprises a turntable and a vertical linear axis. The inspected object is mounted on a fixture on the turntable, while a laser sensor is attached to the vertical axis, see Figure 1. The fixture incorporates high-precision tooling spheres, facilitating the calibration of the sensor-to-fixture relationship and, consequently, the alignment of the workpiece and its key features. These metrology-grade spheres, manufactured to tight tolerances, are strategically positioned to enable robust coordinate system construction and ensure visibility from the sensor across a wide range of workpieces (Lindqvist et al. 2023).

The measurement process begins with the system in its home configuration, where an initial calibration sequence is performed by measuring the tooling spheres. The sensor then conducts dimensional measurements of the visible workpiece features. Whenever the system configuration changes, due to motion of either the turntable or vertical axis, a re-calibration sequence is executed. This involves measuring a set of spheres, with at least three overlapping with the previously measured set, a requirement referred to as the *3-sphere* constraint. This methodology is extendable to a general case with N_S common spheres. Once all features have been measured, the system returns to its home configuration.

Notably, the system does not rely on the mechanical accuracy of its actuators but reconstructs the sensor-to-feature relationship through repeated sphere measurements, ensuring positional consistency across different configurations.

This approach enhances accuracy and enables sensor mounting on a robotic system, which offers greater workspace reachability and faster motion than a coordinate measuring machine (CMM). However, fixture geometry and workpiece occlusions complicate manual identification of feasible system configurations. Optimally determining an optimal sequence of configurations that satisfies the *3-sphere* constraint, while ensuring unobstructed feature visibility, necessitates an automated optimization framework.

3.2. Single measurement planning

Given:

- a configuration (z, θ) for the mechanism carrying the sensor and the features;
- a feature k represented by a point $p_k(\theta) \in \mathbb{R}^{\not\vdash}$ and by its unit normal vector $\hat{n}_k(\theta) \in R^3$;
- the source point of the sensor's laser beam $\lambda(z) \in R^3$,

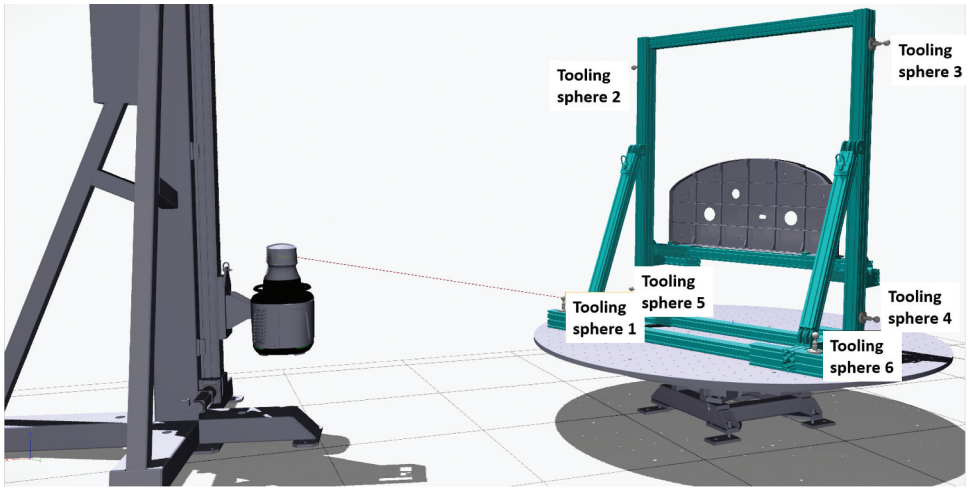

Figure 1. Laser beam pointing towards a tooling sphere.

it is possible to easily compute azimuth and elevation (α, β) such that the laser beam points towards p_k. Moreover, configuration (z, θ) is said to *cover* feature k if and only if the two following criteria are met:

f1) **no feature occlusion**: the laser beam has free sight, i.e. there is no obstacle that intersects the vector $\sigma_k(z, \theta) = \lambda(z) - p_k(\theta)$;

f2) **feature measurement quality**: the angle between the laser beam and n_k is within the tolerated value Δ_k, which ensures a high-qualitative measurement of feature k.

The angle $\delta_k(z, \theta)$ between the laser beam and the normal to feature k, when the mechanism is at configuration (z, θ), can be expressed as $\delta_k = \arccos \frac{\hat{n}_k^T \sigma_k}{\|\sigma_k\|} \leq \Delta_k$, where the dependency to (z, θ) is omitted. In Figure 2, the above defined geometrical entities are depicted.

s1) **no sphere occlusion**: the laser beam pointing at the centre $s_j(\theta)$ of sphere j (excluding the sphere) has free sight. Specifically, with R being the sphere's radius and by defining $l_j(\theta, z) = \lambda(z) - s_j(\theta)$, there should be no surrounding object intersecting the line segment $\left(1 - \frac{R}{\|l_j(\theta,z)\|}\right) l_j(\theta, z)$.

In practice, this test is done after the deletion of the sphere's triangulated model and performing a collision test. Moreover, to account for robustness when running the measurement in reality, six other rays are cast towards the sphere's visible hemisphere: this is done to increase the probability for the sphere of being actually *seen*.[1]

3.3. Configuration space discretization

The configuration space $\mathbb{R} \times \mathbb{S}^1$ (Cartesian product between real numbers and the unit circle) is discretized. The initial strategy is to discretize both DOFs uniformly, by creating N_Z samples for the first DOF and N_θ for the second one. Therefore, one obtains the set $\mathcal{C} = \{1, \ldots, N_Z \times N_\theta\}$, where each element corresponds to a specific configuration.

For each feature k, the set $\mathcal{C}_F(k) \subseteq \mathcal{C}$ of all configurations *covering* feature k according to 1 and 2 can be defined. Conversely, given a configuration $i \in \mathcal{C}$, define the set $\mathcal{F}(i) \subseteq \mathcal{F}$ of all features *covered* by configuration i according to 1 and 2.

For each sphere j, the set $\mathcal{C}_S(j) \subseteq \mathcal{C}$ of all configurations *seeing* sphere j according to 1 can be defined. Conversely, given a configuration $i \in \mathcal{C}$, the set $\mathcal{S}(i) \subseteq \mathcal{S}$ of all spheres *seen* by configuration i according to 1 can also be defined.

Efficient collision queries or ray-casting techniques enable fast computations of criteria 1 and 1, therefore is is possible to retrieve the entire matrix for feature-

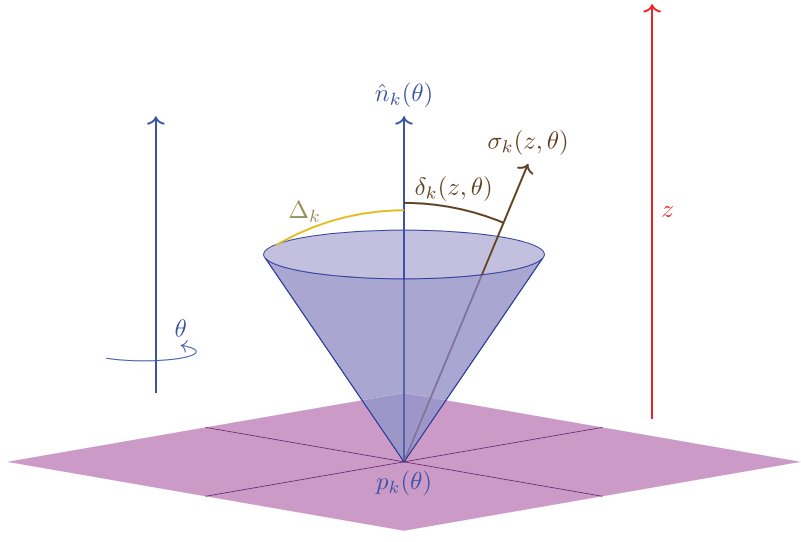

Figure 2. Measurement cone allowed for feature k, placed on a rotating (θ) table. Sensor placed on a linear axis (z). Note that additional information regarding the tooling spheres can be retrieved before carrying out tests 1 and 2 above. Namely, an additional free-sight test is done for each of the tooling spheres at a given mechanism configuration. Indeed, configuration (z, θ) is said to *see* sphere j if and only if the following criterion is met.

configuration feasibility in some seconds, even when CAD models consist of millions of triangles.

Due to the efficiency of such tests, it is possible to generate many candidate sensor positions covering a feature.

These candidates can even be ranked according to their measurement quality, based on the angle δ_k between the laser beam and surface normal (cf. 2). In fact, each pair of sensor position $i \in \mathcal{C}$ and feature $k \in \mathcal{F}$ may be assigned a quality index q_{ik}, based on the incidence angle $\delta_k(z, \theta)$. To include this aspect in the mathematical programming formulation, providing better measurements, one can introduce a weight $w_i, \quad i \in \mathcal{C}$. This weight might be calculated by the sum of the quality indices q_{ik} for all features, *i.e.* $\sum_{k \in \mathcal{F}} q_{ik}$. That leads to an optimization problem where the function to be minimized is the sum of all weights for the chosen configurations. This is currently not done and the authors suggest that as possible future work.

4. Optimization

Optimizing cycle time while fulfilling the measurement criterion 2 and the *3-sphere* constraints depends on the chosen configurations, their order, on the time taken to move the mechanism between them and the measuring time. However, the authors have previously highlighted that each time the mechanism configuration is changed, then a quite slow calibration process needs to be carried out. Moreover, the laser sensor is very fast in changing its azimuth and elevation. Therefore, a good approximation for the total measurement time is proportional to the number of configurations chosen, i.e. the number of times the vertical axis or the turnable move and a new calibration takes place. In the rest of the paper, this is assumed to be the main objective.

An overview of the proposed method is presented in Figure 3. Each block in it is explained in its own subsection.

4.1. Preprocessing

The mechanism configurations are pruned in order to remove those having both the feature and sphere sets included in the corresponding sets of another configuration. Basically, it is possible to remove any configuration $i \in \mathcal{C}$ if $\exists \bar{i} \in \mathcal{C}, \bar{i} \neq i$ such that

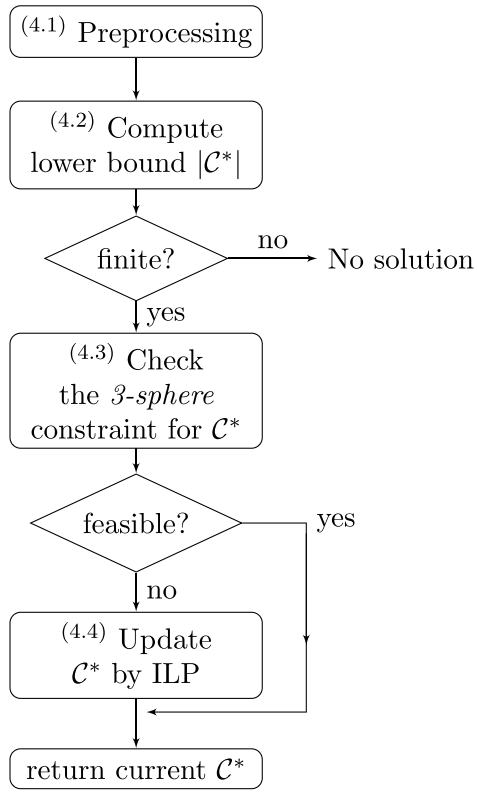

Figure 3. Flowchart for sensor position optimization.

$\mathcal{F}(i) \subseteq \mathcal{F}(\bar{i}) \wedge \mathcal{S}(i) \subseteq \mathcal{S}(\bar{i})$. In practice, the removed configurations are often close to each other in the configuration space (z, θ).

If distances between configurations need to be evaluated, for example, when explicitly modeling cycle time in sequencing problems, then one should reconsider all configurations again.

4.2. A lower bound on the number of configurations needed

When preprocessing is done then an updated set $\mathcal{C}$ of configurations is computed. The initial question is whether there is a solution at all and in its lower bound.

To do that, one can disregard the *3-sphere* constraint, and minimize the number of configurations needed to measure all features. This simplification leads to a classic set covering problem (Rubin 1973):

$$\min \sum_{i \in C} x_i \tag{1a}$$

$$s.t. \sum_{i \in C_F(k)} x_i \geq 1, \quad k \in \mathcal{F}, \tag{1b}$$

$$x_i \in \{0,1\},\ i \in C \tag{1c}$$

The unknowns x_i take the value 1 if and only if configuration i is chosen. Constraints (4.2) impose that there has to be at least one chosen configuration *seeing* feature k; this should hold for all features in $\mathcal{F}$.

The optimal value obtained can be used as a lower bound for the minimum number of configurations needed when the *3-sphere* constraint is added. It can also be used as a potential solution to the entire problem, after the verification of its feasibility w.r.t the same constraint. In the following section a method to verify that is provided.

4.3. Feasibility of a given configuration set

After the computation of the lower bound, one may verify whether there is a given order for the obtained configurations, satisfying the *3-sphere* constraint. This can be done with the algorithm below.

Due to the need to model sequences algorithms solving the TSP can be adapted: one of them is the dynamic programming approach (Held and Karp 1962) the results for a given instance with. This algorithm has the advantage to be able to model and optimize also some cost/time between mechanism configurations, compared to model (1), in addition to finding a feasible set of configurations. A function ϕ is optimized for all configurations except the initial one, i.e. for all $i \in \mathcal{C}^1 = \mathcal{C}\backslash\{1\}$ (lines 2–3). In the TSP, this function usually models the cost of a sub-tour terminating at a node i and visiting all nodes in A. To simplify notation, let us define $A^i = A\backslash\{i\}$ and denote the cardinality operator on a set A with $|A|$.

Algorithm 1. Modified Held-Karp

1: $\mathcal{C}^1 = \{2,3,\ldots,|\mathcal{C}|\}$
2: **for all** $i \in \mathcal{C}^1$ **do**
3: $\quad \phi(\{i\},i) \leftarrow d(1,i)$
4: **end for**
5: **for all** $l \in \mathcal{C}^1$ **do**
6: $\quad$ **for all** $A \subseteq \mathcal{C}^1, |A| = l$ **do**
7: $\quad\quad$ **for all** $i \in A$ **do**
8: $\quad\quad\quad \phi(A,i) = \min_{k \in A^i}[\phi(A^i,k) + d(k,i)] + p(A^i,i)$
9: $\quad\quad$ **end for**
10: $\quad$ **end for**
11: **end for**
12: opt $= \min_{i \in \mathcal{C}^1}\left[\phi(\mathcal{C}^1,i) + d(i,1)\right]$

The optimal value 'opt' is given by choosing the minimal value to return to the initial configuration 1. Note that line 8 includes an additional term p, which basically stops the construction of the dynamic programming graph due to the *3-sphere* constraint not being fulfilled. Indeed, p is defined in the following way:

$$p(\mathcal{A}^i,i) = \begin{cases} \infty, & \text{if } \left|\mathcal{S}(i) \cap \bigcup_{k \in A^i} \mathcal{S}(k)\right| < 3 \\ 0, & \text{otherwise.} \end{cases}$$

If such an order of configurations is found, then the solution is optimal also w.r.t. *3-sphere* constraint.

Note that this approach is restricted to problems with $|\mathcal{C}|$ lower than about 20, due to memory limits of the current computers. In larger cases, feasibility can be verified by a simple constructive algorithm, see Appendix B.

When optimality cannot be proven, an optimization model which captures the sequential nature of the problem to find an optimal solution is formulated.

4.4. An exact approach

Here, the authors present the main contribution of the article, namely a model to handle the sequential nature of the problem, which is not caught by the set covering model.

To do that, let us introduce a *period* concept, modeling a time slot in which some configurations may be chosen (or unblocked), due the *3-sphere* constraint being satisfied given the previously chosen configurations. The resulting Integer Linear Programming (ILP) model contains unknowns x_{it}, taking the value 1 if and only if configuration i is active in the time slot t and is the following:

$$\min \sum_{i \in C} x_i |\mathcal{T}| \tag{2a}$$

$$s.t. \sum_{i \in C_F(k)} x_i |\mathcal{T}| \geq 1, \quad k \in \mathcal{F}, \tag{2b}$$

$$\sum_{j \in s(i)} s_{j(t-1)} \geq N_{S^{x_{it}}},\ i \in C,\ t \in \mathcal{T}^1, \tag{2c}$$

$$|C_S(j)| \cdot s_{jt} \geq \sum_{i \in C_{s(j)}} x_{it},\ j \in S,\ t \in \mathcal{T}, \tag{2d}$$

$$s_{jt} \leq \sum_{i \in C_{s(j)}} x_{it}, \quad j \in S,\ t \in \mathcal{T}, \tag{2e}$$

$$x_{i(t-1)} \leq x_{it}, \quad i \in C,\ t \in \mathcal{T}^1, \tag{2f}$$

$$s_{j(t-1)} \leq s_{jt}, \qquad j \in S,\ t \in \mathcal{T}^1, \tag{2g}$$

$$\sum_{i \in C} x_{i1} = 1, \tag{2h}$$

$$x_{it} \in \{0,1\}, \qquad i \in C,\ t \in \mathcal{T}, \tag{2i}$$

$$s_{jt} \in \{0,1\}, \qquad j \in S,\ t \in \mathcal{T}, \tag{2j}$$

Note that, after a configuration has been activated at a given time slot, it will be active for all successive periods: this is modeled by (2f). The same logic holds for the sphere variables s_{jt}: they take the value 1 if and only if sphere j is active in the time slot t, remaining active until the end, due to (2 g). This model will be named 'Active After' due to this constraint, to differentiate it from another model (see Appendix 7), where variables are one only during the activation step and return to 0 afterwards, which is the reason why it is has been named 'ActiveOnce'. The objective function, therefore, needs to consider only the last time period $|\mathcal{T}|$, to be able to count the number of chosen configurations, (2a). Constraints (2b) are the same as (4.2), with the difference that they only yield the last time period $|\mathcal{T}|$. Constraints (2c) impose that configuration i may be active at period t only if at least N_S spheres in $\mathcal{S}(i)$ have been active in the previous period, $t-1$. Similarly, sphere j may not be active if there is no active configuration in $\mathcal{C}_S(j)$ in the same time slot, (2e). Moreover, if one configuration in $\mathcal{C}_S(j)$ is active, then sphere j is activated, (2d). Last constraint, (2 h), simply states that only one configuration may be active in the first time slot. The set $\mathcal{T}^1$ is defined as $\mathcal{T} \backslash \{1\}$.

Notations for the different sets, variables and constants are summarized in Tables 1 and 2.

Note that the number of time slots $|\mathcal{T}|$ is a design parameter. A strategy often adopted has been to determine it by checking the lower bound, obtained in Section 4.2. Then, one can start setting it equal to the lower bound and increase it until a solution is found. Different strategies are possible here. The authors have chosen the initial number of time slots as the $\min(|\mathcal{C}^*|, |\mathcal{S}| + 1 - N_S)$, since adding only one sphere at the time constitutes the worst case scenario and the lower bound ($|\mathcal{C}^*|$) is often a good guess.

Table 3 illustrates the solution of a fictitious instance with 4 time slots: in the first one configuration x_{32} is chosen, which lets the laser sensor *see* the three spheres s_1, s_4, s_6, covering only 10% of the features to be measured. In the second time slot three configurations are chosen (x_2, x_7, x_{42}), which can *see* s_9 in addition to the previous spheres, covering a total of 50% of the features, and so on. Note that it is not given in which order the mechanism assumes these configurations, but they are all visited before moving to the next time period.

Table 1. Useful sets.

Notation	Set of indices of
$\mathcal{C}$	mechanism configurations
$\mathcal{F}$	inspection features
$\mathcal{S}$	tooling spheres
$\mathcal{C}_S(j)$	mechanism configurations for the sensor *seeing* sphere j
$\mathcal{C}_F(k)$	mechanism configurations for the sensor *covering* feature k
$\mathcal{F}(i)$	features *covered* by the sensor at mechanism configuration i
$\mathcal{S}(i)$	spheres *seen* by the sensor at mechanism configuration i
$\mathcal{T}$	time slots
$\mathcal{T}^1$	time slots without the first one

Table 2. Constants and variables.

Notation	Description
N_S	minimum number of spheres required to be in common between $\mathcal{S}(i_1)$ and $\mathcal{S}(i_2)$ for two consecutive mechanism configurations i_1, i_2
$\lvert * \rvert$	cardinality of a set *
x_{it}	decision variable indicating that mechanism configuration i is active at time slot t if one, not active otherwise
s_{jt}	decision variable indicating that sphere j is active at time slot t if one, not active otherwise

Table 3. An artificial case showing the configurations and spheres activated at each time slot, the accumulated active ones, and the features covered by the accumulated active configurations.

Slots		Configurations	Spheres	Covering
$t=1$	New	x_{32}	s_1, s_4, s_6	
	Active	x_{32}	s_1, s_4, s_6	
$t=2$	New	x_2, x_7, x_{42}	s_9	
	Active	x_2, x_7, x_{32}, x_{42}	s_1, s_4, s_6, s_9	
$t=3$	New	x_{16}	s_{12}, s_{13}	
	Active	$x_2, x_7, x_{16}, x_{32}, x_{42}$	$s_1, s_4, s_6, s_9, s_{12}, s_{13}$	
$t=4$	New	x_5, x_{61}	s_2	
	Active	$x_2, x_5, x_7, x_{32}, x_{42}, x_{61}$	$s_1, s_2, s_4, s_6, s_9, s_{12}, s_{13}$	

5. Computational results and simulation

These algorithmS have been implemented in C + + and integrated them with the Industrial Path Solutions (IPS) software for the simulation of kinematics, collision detection and for problem setup. The pilot case consists of an artefact with hundreds of features to be measured (defined by a point and a normal), with spheres of radius 19 mm. The system is illustrated in Figure 4, both in the real world and the simulation software.

The computational studies were carried out on a desktop PC running Windows 10 on a AMD Ryzen 7 2700X 8-Core, 3700 MHz. The number of features $|\mathcal{C}|$ varies, together with the discretization resolution of the DOFs for each axis, while fixing the number of tooling spheres $|\mathcal{S}|$ to 6. Indeed, the fixture is often manufactured together with the tooling spheres and allows several products to be inspected after each other. The input consists of triangulated models (imported in different CAD formats), with about 700 thousand triangles.

A preliminary analysis with an industrial case shows that the preprocessing step, Section 4.1, is very powerful. Indeed, in Figure 5 it is possible to note how the number of potential configurations grows almost linearly when increasing the sampling of each of the two axes, in contrast to a nominal quadratic growth.

The first study regards the limits of the solver for the proposed models. Test instances were generated by modifying a problem with 776 features from a real manufactured prototype. Half of them were selected as basis to produce random instances: for each fixed number of additional features (233, 388, 543), 10 random instances were created. In Figure 6 one can see the median and mean times for solving both proposed models, (2) and (3): the 'ActiveAfter' model is

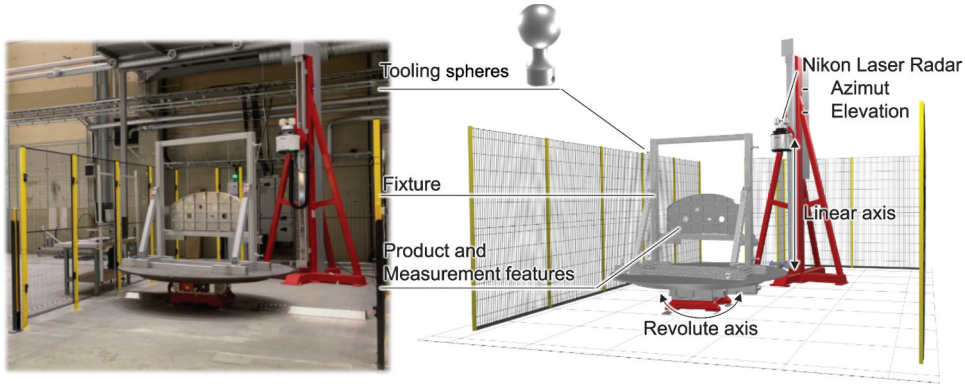

Figure 4. Pilot case from Saab Aeronautics. On the left the real cell; on the right the digital model in IPS.

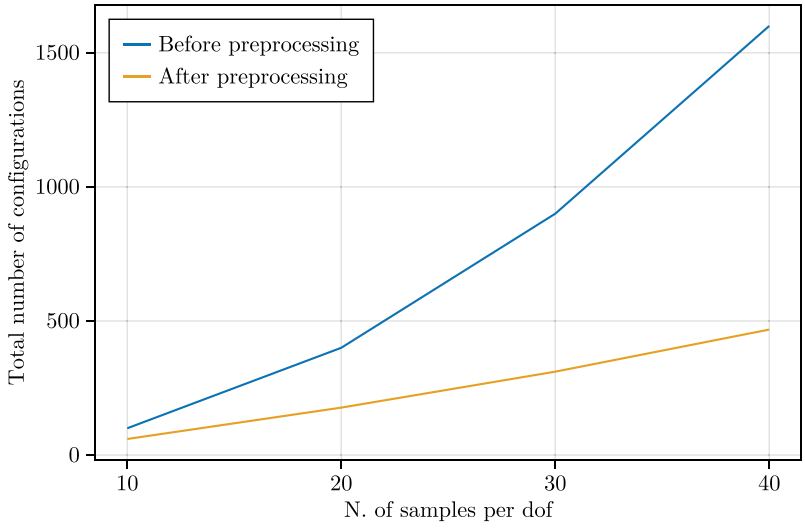

Figure 5. Pre-processing effect on the number of representative configurations.

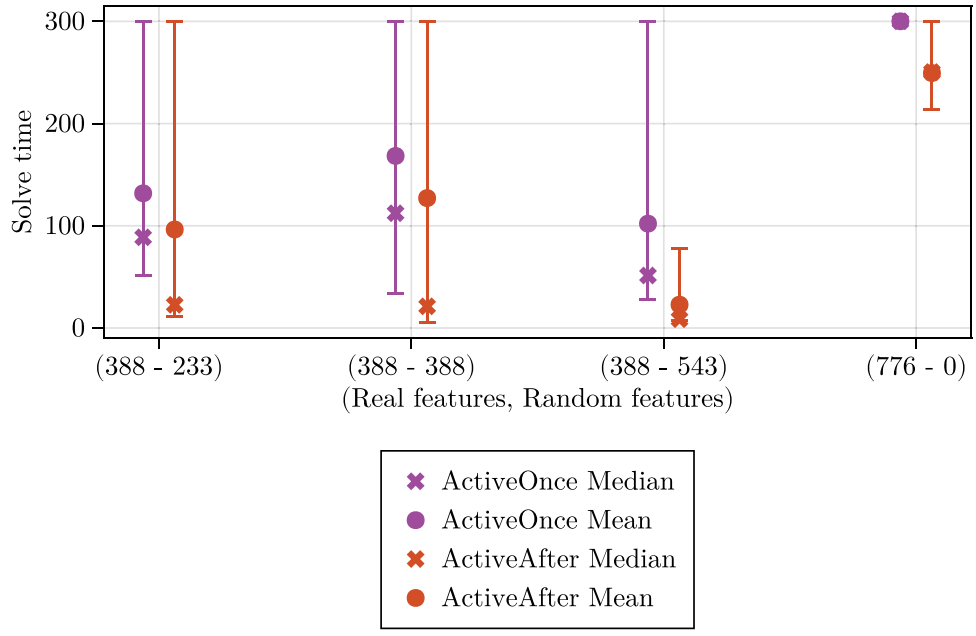

Figure 6. Computing times (seconds) when varying the number of features.

faster than 'ActiveOnce' and the time limit of 5 minutes is approached in the real case having 776 features.

An interesting question is how one should choose the discretization of the DOFs, cfr. Section 3.3, due to the trade-off between computational time and solution quality (measurement time). The number of configurations $|\mathcal{F}|$ was varied on the same instance and solved, together with the computation of the lower bound by the set covering model. The ILP solver used was HiGHS (Huangfu and Hall 2018).

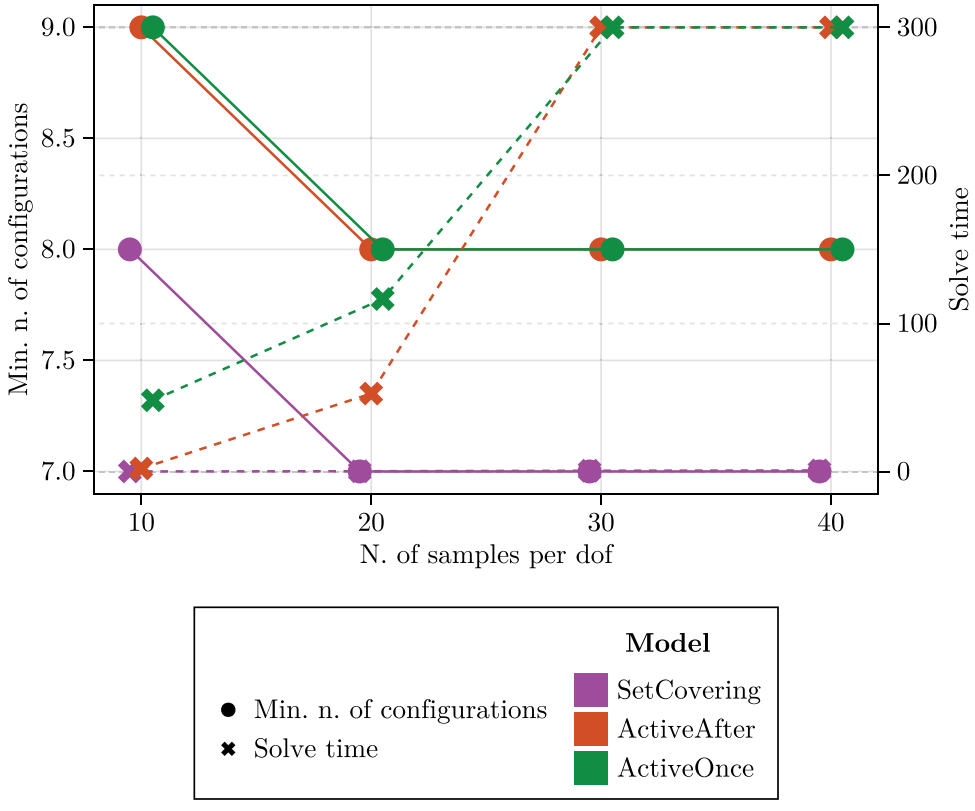

Figure 7. Number of configurations needed vs computing time (seconds) for one representative instance solved by the two presented ILP models "ActiveAfter" and "ActiveOnce" and by its "SetCovering" relaxation.

The graph in Figure 7 presents the results for a given instance with 388 real features and 233 random ones. The graph pattern is common with most of all other instances. It is possible to note how the *3-sphere* constraint makes the problem more difficult to solve w.r.t. its set covering relaxation. Moreover, its lower bound is quite tight. The minimal number of configurations, being the objective value, is directly proportional to the measurement time (or cycle time). A remark to be done is that when the time limit is reached, the best solution so far found by the solver is reported. It may be noted that the cycle time decreases when switching from 10 to 20 samples per DOF, remaining constant afterwards. This sampling strategy might be used as a 'rule of thumb' in adopting this tool in practice.

Another trade-off that the inspection engineer has to deal with is between cycle time and accuracy. It is obvious that by increasing the maximum allowed angle Δ between the laser beam and the feature normal, it should be possible to measure the same feature by more laser radar positions. Consequently, there is a higher probability that the minimum number of laser radar positions needed decreases. The experiment was designed by fixing Δ and creating ten random instances. In Figure 8 the median number of optimal configurations is illustrated together with the entire interval of optima obtained. It is possible to see that the improvement in measurement time (or cycle time) is substantial when going from 30 to 45 degrees. Also here, note that the cycle time is directly proportional to the number of configurations chosen by the algorithm. Of course, it is up to the engineers deciding whether the potential degradation in the measurement accuracy is worth the gain in cycle time.

Note that, being a newly developed system (more extensively described in (Lindqvist et al. 2023)), there is no data accounting for how previous measurements programs were done. Indeed, the inspection engineers found it difficult to create feasible programs at all, since no automatic software existed.

5.1. Industrial test case

A complete inspection program has been generated by using the algorithms proposed in this article, with the help of a simulation software. The setup is the one described in the article with a total of 301 features to be measured: 212 surface points, 76 cylinders and 13 planes, see Figure 9.

The first operation consists in creating N_θ positions for the turntable and N_Z for the vertical track: they are uniformly sampled every 10 degrees, respectively, every 10 mm, giving a total of 10,767 configurations. When this is done each feature is planned to be measured by a sensor configuration, satisfying constraints 1, 2, and 1. This step takes about 10 seconds.

Afterwards, 15 configurations are obtained by the proposed algorithm. Their optimal order is computed by running the algorithm in Section 4.3, where the

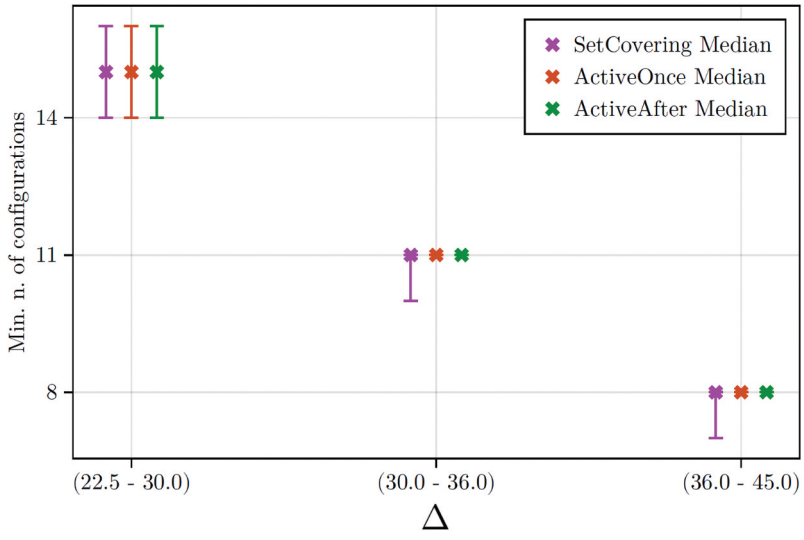

Figure 8. Number of configurations needed when varying Δ (degrees).

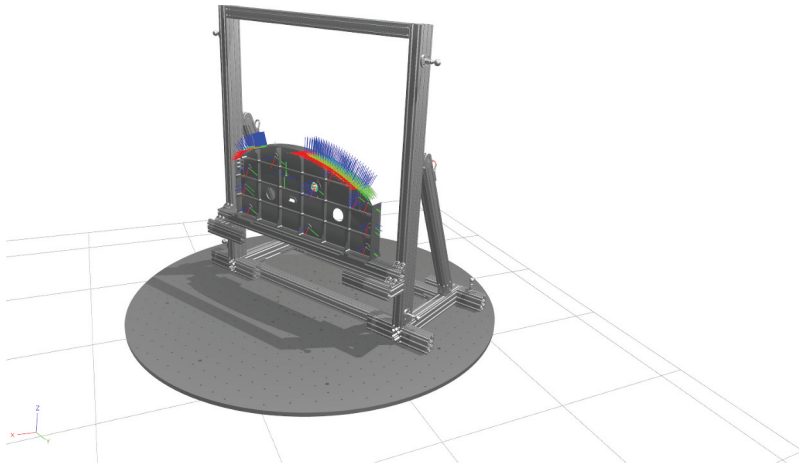

Figure 9. Workpiece investigated in the pilot case.

distance $d(k,i)$ between configurations (θ_k, z_k) and (θ_i, z_i) is given by the time it takes to move between them. This is computed as $\max(|\theta_k - \theta_i|/\omega, |z_k - z_i|/v)$, being ω the angular velocity of the turntable and v the linear speed of the vertical axis.

The optimal sequence is listed in Figure 10, together with the number of features measured and the tooling spheres seen.

5.2. Results from real implementation work at saab aeronautics

At Saab Aeronautics, the developed algorithm for managing the *3-sphere* criterion has been successfully integrated into offline dimensional metrology programming. The results demonstrate compliance with Saab Aeronautics' objectives, achieving significant reductions in offline programming time while maintaining measurement quality. The estimated reduction in physical measurement time ranges between 5% and 20%.

Observations from the *T-7* project, containing 187–243 dimensional features, indicated that about 90% reduction in total offline programming time was achieved, compared to manual preparation and direct online teaching with laser radar. The complexity of manual preparation becomes particularly evident in the absence of automated measurement systems, in which case the manual workflow would have been even more intricate, increasing the cognitive load on measurement engineers.

Conf.	Vertical track (mm)	Turntable (deg.)	N. features	Tooling spheres					
				1	2	3	4	5	6
1	0	-180	4	x		x	x	x	x
2	140	-110	10	x	x	x	x		x
3	200	-90	5	x	x	x			x
4	210	160	6			x	x	x	x
5	180	160	1			x	x	x	x
6	0	90	8	x	x	x	x	x	x
7	320	-100	8	x	x	x			x
8	0	170	6			x	x	x	x
9	2580	90	83		x	x	x		
10	1210	120	30			x	x	x	x
11	1190	-20	17	x	x	x			
12	2030	-10	75	x		x	x		
13	0	20	35	x	x	x	x	x	
14	2570	90	1		x	x	x		
15	2490	90	12		x	x	x		

Figure 10. Sequence of configurations for the optimal inspection program.

6. Conclusion

In this work, the authors have investigated a laser radar inspection system where a re-calibration constraint during measurements makes even finding a feasible program a difficult task to be carried out manually. A customized model and algorithm have been designed, showing that it is possible to optimize inspection programs off-line, with hundreds of features, considering as goal the measurement/cycle time.

The computational challenges regarding how large instances one can solve in a given time open up for further development. Among the promising ideas identified, there is the introduction of non-uniform sampling for the DOFs and a lazy generation of good configurations based on geometrical characteristics. Moreover, choosing the most promising configurations based on several measurement quality criteria might be very useful as well.

The provided algorithms may be used in a software tool to automatically generate optimal programs or as a support for the inspection engineers who must deal with the trade-offs between: measurement accuracy vs. cycle time on one hand, and between the quality of the final inspection program vs. the man-hours invested to generate it.

Note

1. A worst case scenario might occur when an occluding object, with a small hole in it, lets the laser go through it, but only at the nominal position.

Acknowledgments

The authors would like to thank the colleagues at the Fraunhofer-Chalmers Centre, in particular, Dr. Robert Bohlin and Dr. Johan S. Carlson, and are grateful to the colleagues at Saab Aeronautics, for useful discussions and help in the realization of this work. This work benefited from funding from the Swedish Innovation Agency Vinnova, as part of its Digi-Q research project. It is also part of the Sustainable Production Initiative (SPI) and the Production Area of Advance at Chalmers University of Technology.

During the preparation of this work, the authors used ChatGPT 4o, in order to improve readability and correct grammatical errors. After using this tool, the authors rigorously reviewed and edited the content as needed and take full responsibility for the content of the publication.

Disclosure statement

No potential conflict of interest was reported by the author(s). Certain commercial systems and software identified in this paper do not imply recommendation or endorsement by FCC, Saab or by the authors, nor does it imply that the products identified are necessarily the best available for the purpose. This holds even if the authors might have employment and/or ownership relationship with the software organization.

References

Archenti, A., W. Gao, A. Donmez, E. Savio, and N. Irino. 2024. "Integrated Metrology for Advanced Manufacturing." *CIRP Annals* 73 (2): 639–665. https://doi.org/10.1016/j.cirp.2024.05.003.

Azamfirei, V., F. Psarommatis, and Y. Lagrosen. 2023. "Application of Automation for In-Line Quality Inspection, a Zero-Defect Manufacturing Approach." *Journal of Manufacturing Systems* 67:1–22. https://doi.org/10.1016/j.jmsy.2022.12.010.

Bircher, A., K. Alexis, M. Burri, P. Oettershagen, S. Omari, T. Mantel, and R. Siegwart. 2015. "Structural Inspection Path Planning via Iterative Viewpoint Resampling with Application to Aerial Robotics." *2015 IEEE International Conference on Robotics and Automation (ICRA)*, Seattle, WA, USA, 6423–6430.

Bogaerts, B., S. Sels, S. Vanlanduit, and R. Penne. 2018. "A Gradient-Based Inspection Path Optimization Approach." *IEEE Robotics and Automation Letters* 3 (3): 2646–2653. https://doi.org/10.1109/LRA.2018.2827161.

Dowling, M. M., P. M. Griffin, K.-L. Tsui, and C. Zhou. 1997. "Statistical Issues in Geometric Feature Inspection Using Coordinate Measuring Machines." *Technometrics* 39 (1): 3–17. Accessed 2023-05-12. http://www.jstor.org/stable/1270764.

Fan, X., L. Zhang, B. Brown, and S. Rusinkiewicz. 2016. "Automated View and Path Planning for Scalable Multi-Object 3D Scanning." *ACM Transactions on Graphics* 35 (6): 1–13. https://doi.org/10.1145/2980179.2980225.

Galceran, E., and M. Carreras. 2013. "A Survey on Coverage Path Planning for Robotics." *Robotics and Autonomous Systems* 61 (12): 1258–1276. https://doi.org/10.1016/j.robot.2013.09.004.

Glorieux, E., and P. Franciosa. 2020. "Coverage Path Planning with Targetted Viewpoint Sampling for Robotic Free-Form Surface Inspection." *Robotics and Computer-Integrated Manufacturing* 61:61. https://doi.org/10.1016/j.rcim.2019.101843.

Gospodnetić, P., D. Mosbach, M. Rauhut, and H. Hagen. 2022. "Viewpoint Placement for Inspection Planning." *Machine Vision and Applications* 33 (1): 1–21. https://doi.org/10.1007/s00138-021-01252-z.

Held, M., and R. M. Karp. 1962. "A Dynamic Programming Approach to Sequencing Problems." *Journal of the Society*

for Industrial and Applied Mathematics 10 (1): 196–210. https://doi.org/10.1137/0110015.

Huangfu, Q., and J. A. Julian Hall. 2018. "Parallelizing the Dual Revised Simplex Method." *Mathematical Programming Computation* 10 (1): 119–142. https://doi.org/10.1007/s12532-017-0130-5.

Jing, W., D. Deng, Y. Wu, and K. Shimada. 2020. "Multi-Uav Coverage Path Planning for the Inspection of Large and Complex Structures." *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 1480–1486. IEEE.

Jose, K., and D. Kumar Pratihar. 2016. "Task Allocation and Collision-Free Path Planning of Centralized Multi-Robots System for Industrial Plant Inspection Using Heuristic Methods." *Robotics and Autonomous Systems* 80:34–42. https://doi.org/10.1016/j.robot.2016.02.003.

Lattanzi, D., and G. Miller. 2017. "Review of Robotic Infrastructure Inspection Systems." *Journal of Infrastructure Systems* 23 (3): 04017004. https://doi.org/10.1061/(ASCE)IS.1943-555X.0000353.

Lauri, M., J. Pajarinen, J. Peters, and S. Frintrop. 2020. "Multi-sensor next-best-view planning as matroid-constrained submodular maximization." *IEEE Robotics and Automation Letters* 5 (4): 5323–5330. https://doi.org/10.1109/LRA.2020.3007445.

Li, B., P. Feng, L. Zeng, C. Xu, and J. Zhang. 2018. "Path Planning Method for On-Machine Inspection of Aerospace Structures Based on Adjacent Feature Graph." *Robotics and Computer-Integrated Manufacturing* 54:17–34. https://doi.org/10.1016/j.rcim.2018.05.006.

Lindqvist, R. P., D. Strand, M. Nilsson, V. Collins, J. Torstensson, J. Kressin, D. Spensieri, and A. Archenti. 2023. "3D Model-Based Large-Volume Metrology Supporting Smart Manufacturing and Digital Twin Concepts." *Metrology* 3 (1): 29–64. https://doi.org/10.3390/metrology3010002.

Liu, Y., W. Zhao, H. Liu, Y. Wang, and X. Yue. 2022. "Coverage Path Planning for Robotic Quality Inspection with Control on Measurement Uncertainty." *IEEE/ASME Transactions on Mechatronics* 27 (5): 3482–3493. https://doi.org/10.1109/TMECH.2022.3142756.

Mahmud, M., D. Joannic, M. Roy, A. Isheil, and J.-F. Fontaine. 2011. "3D Part Inspection Path Planning of a Laser Scanner with Control on the Uncertainty." *Computer-Aided Design* 43 (4): 345–355. https://doi.org/10.1016/j.cad.2010.12.014.

Pan, S., H. Hu, and H. Wei. 2022. "Scvp: Learning One-Shot View Planning via Set Covering for Unknown Object Reconstruction." *IEEE Robotics and Automation Letters* 7 (2): 1463–1470. https://doi.org/10.1109/LRA.2022.3140449.

Raffaeli, R., M. Mengoni, M. Germani, and F. Mandorli. 2013. "Off-Line View Planning for the Inspection of Mechanical Parts." *International Journal on Interactive Design & Manufacturing (IjideM)* 7 (1): 1–12. https://doi.org/10.1007/s12008-012-0160-1.

Rubin, J. 1973. "A Technique for the Solution of Massive Set Covering Problems, with Application to Airline Crew Scheduling." *Transportation Science* 7 (1): 34–48. https://doi.org/10.1287/trsc.7.1.34.

Sadaoui, S. E., C. Mehdi-Souzani, and C. Lartigue. 2018. "Computer-Aided Inspection Planning: A Multisensor High-Level Inspection Planning Strategy." *Journal of Computing and Information Science in Engineering* 19 (2). https://doi.org/10.1115/1.4041970.

Sadaoui, S. E., C. Mehdi-Souzani, C. Lartigue, and M. Brahim. 2022. "Automatic Path Planning for High Performance Measurement by Laser Plane Sensors." *Optics and Lasers in Engineering* 159:107194. https://doi.org/10.1016/j.optlaseng.2022.107194.

Salman, R., J. S. Carlson, F. Ekstedt, D. Spensieri, J. Torstensson, and R. Söderberg. 2016. "An Industrially Validated CMM Inspection Process with Sequence Constraints." *Procedia CIRP* 44, 138–143. 6th CIRP Conference on Assembly Technologies and Systems (CATS), https://doi.org/10.1016/j.procir.2016.02.136.

Scott, W. R., G. Roth, and J.-F. Rivest. 2001. "View Planning with a Registration Constraint." *Proceedings Third International Conference on 3-D Digital Imaging and Modeling*, 127–134.

Sheng, W., N. Xi, J. Tan, M. Song, and Y. Chen. 2003. "Viewpoint Reduction in Vision Sensor Planning for Dimensional Inspection." *IEEE International Conference on Robotics, Intelligent Systems and Signal Processing, 2003. Proceedings*, Vol. 1, 249–254. IEEE.

Spensieri, D., E. Ablad, R. Salman, and J. S. Carlson. 2025. "A Unified Sampling Method for Optimal Feature Coverage and Robot Placement." *Robotics and Computer-Integrated Manufacturing* 93:102932. https://doi.org/10.1016/j.rcim.2024.102932.

Yau, H.-T., and C.-H. Menq. 1991. "Path Planning for Automated Dimensional Inspection Using Coordinate Measuring Machines." *Proceedings. 1991 IEEE International Conference on Robotics and Automation*, 1934–1935. IEEE Computer Society.

Zhao, F., X. Xu, and S. Q. Xie. 2009. "Computer-Aided Inspection Planning—The State of the Art." *Computers in Industry* 60 (7): 453–466. https://doi.org/10.1016/j.compind.2009.02.002.

Appendices

Appendix A "ActiveOnce" model

$$\min \sum_{i\in C}\sum_{t\in\mathcal{T}} x_{it} \tag{3a}$$

$$s.t. \quad \sum_{i\in C_F(k)}\sum_{t\in\mathcal{T}} x_{it} \geq 1, \quad k \in \mathcal{F} \tag{3b}$$

$$\sum_{i\in C} x_{it} \leq 1, \quad t \in \mathcal{T}, \tag{3c}$$

$$\sum_{t\in\mathcal{T}} s_{jt} \leq 1, \quad j \in S, \tag{3d}$$

$$\sum_{p=1}^{t-1}\sum_{j\in S(i)} s_{jp} \geq N_{s^{x_{it}}}, \; i \in C, t \in \mathcal{T}^1, \tag{3e}$$

$$\sum_{i\in C_S(j)} x_{it} \geq s_{jt}, \quad j \in S, t \in \mathcal{T}, \tag{3f}$$

$$\sum_{i\in C} x_{i1} = 1 \quad i \in C, \tag{3g}$$

$$x_{it} \in \{0,1\}, \quad i \in C, t \in \mathcal{T} \tag{3h}$$

$$s_{jt} \in \{0,1\}, \quad j \in S, t \in \mathcal{T} \tag{3i}$$

A slightly different way to model the problem in 4.4 is to set x_{it} to 1, only for the time slot it is activated and then let it be 0 again. So there can be maximum one x_{it} active among all $t \in \mathcal{T}$, see (3c). The same logic holds for the sphere variables s_{jt}, see 7. This model is named "ActiveOnce" for that reason.

The objective function, therefore, needs to consider all time periods $|\mathcal{T}|$, to be able to count the number of chosen configurations (3a). The other constraints are updated accordingly.

Appendix B - Feasibility check

The feasibility of a given configuration set may be determined by a customized reduction algorithm: iteratively tracking the included configurations and the union of their sphere sets. Configurations are incrementally included if they share three spheres with the union of the already included sphere sets. The full procedure is described in Algorithm 2.

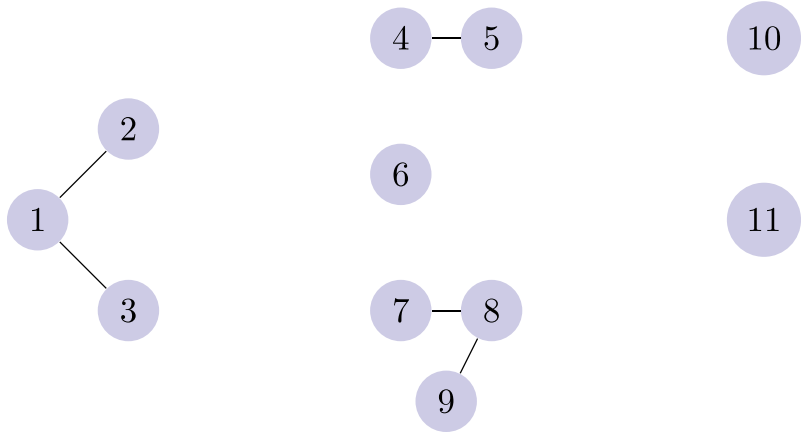

Figure A1. Initial graph.

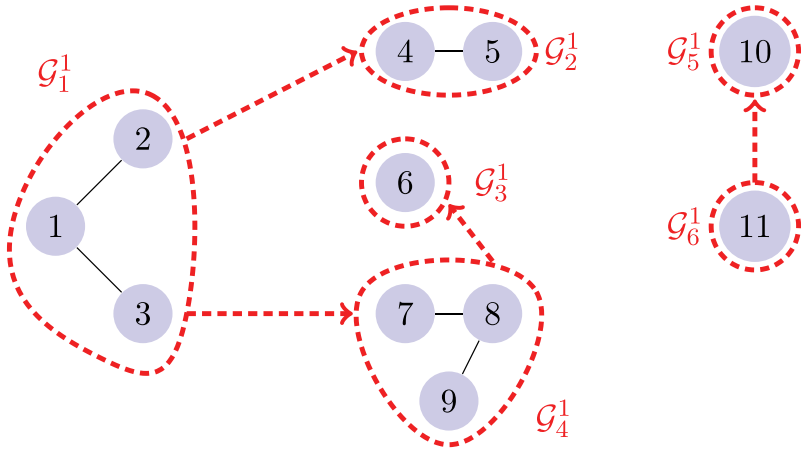

Figure A2. First step reducing the graph.

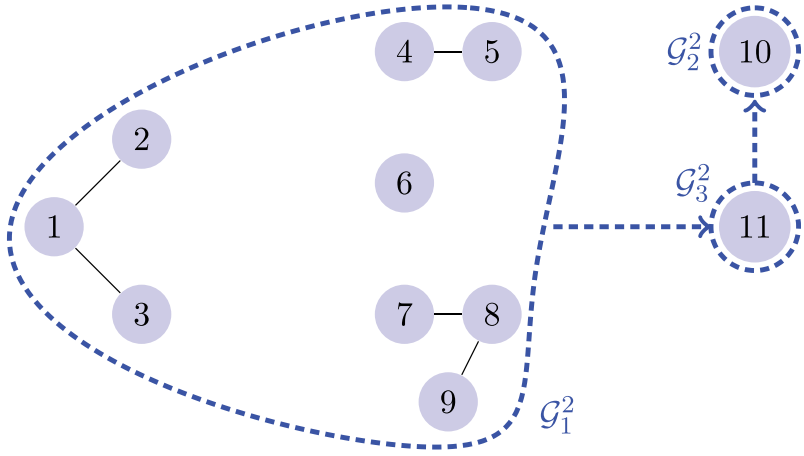

Figure A3. Second step reducing the graph.

Assume the initial configuration has index 1 and let $\mathcal{I} = \{1\}$ denote the set of included configurations (line 1). Then, one can successively add configurations to $\mathcal{I}$, if they satisfy the *3-sphere* constraint, (lines 4-7). The main loop stops successfully when all features have been covered or unsuccessfully, when the set $\mathcal{I}$ of included configurations is the entire $\mathcal{C}$, (line 2). In Figure A1 a graph modeling is given, where nodes represent configurations and edges between nodes modeling configurations i_1 and i_2 are added if they *see* three common spheres. Connected nodes represent clusters, see Figure A2, and can help making the algorithm more efficient: in fact one can add an edge between cluster $\mathcal{G}_1$ and $\mathcal{G}_2$ if there exists a node $i_2 \in \mathcal{G}_2$ having three spheres in common with all spheres seen in cluster $\mathcal{G}_1$, i.e. $\left|\mathcal{S}(i_2) \cap \bigcup_{i \in \mathcal{G}_1} \mathcal{S}(i)\right| \geq 3$. This means that all configurations in a cluster may be added, without needing to check all of them. Figure A3 shows next step for the reduction algorithm.

Algorithm 2. Configuration reduction

1: $\mathcal{I} \leftarrow \{1\}$ ▷Included configurations
2: **while** $\bigcup_{i \in \mathcal{I}} \mathcal{F}(i) \mathcal{F} \vee \mathcal{IC}$ **do**
3: $\mathcal{N} \leftarrow \emptyset$ ▷Configurations to-be included
4: **for** $j \in \mathcal{C} - \mathcal{I}$ **do**
5: **if** $\left|\mathcal{S}(j) \cap \bigcup_{i \in \mathcal{I}} \mathcal{S}(i)\right| \geq 3$ **then**
6: $\mathcal{N} \leftarrow \mathcal{N} \cup \{j\}$
7: **end if**
8: **end for**
9: $\mathcal{I} \leftarrow \mathcal{I} \cup \mathcal{N}$
10: **end while**
11: **if** $\bigcup_{i \in \mathcal{I}} \mathcal{F}(i) = \mathcal{F}$ **then**
12: *return* $\mathcal{I}$
13: **else**
14: *return* $\emptyset$
15: **end if**