Manuscript submitted to International Journal of Vehicle Design

Design Optimization of a Hybrid Electric Vehicle Cooling System Considering Performance and Packaging

Kwang Jae Lee[1]
Department of Mechanical Engineering, University of Michigan,
2350 Hayward Street, Ann Arbor, MI 48109, USA
kwangjae@umich.edu

Namwoo Kang*
The Cho Chun Shik Graduate School of Green Transportation,
Korea Advanced Institute of Science and Technology,
193 Munji-ro, Yuseong-gu, Daejeon 34141, Korea
nwkang@kaist.ac.kr

Michael Kokkolaras
Department of Mechanical Engineering, McGill University
817 Sherbrooke Street West, Montreal, QC H3A 0C3, Canada
michael.kokkolaras@mcgill.ca

Panos Y. Papalambros
Department of Mechanical Engineering, University of Michigan,
2350 Hayward Street, Ann Arbor, MI 48109, USA
pyp@umich.edu

[1]Author's current affiliation is Samsung Electronics.

*corresponding author

Biographical Statements

Kwang Jae Lee received his PhD in Mechanical Engineering from the University of Michigan. He currently works at Samsung Electronics. His research interests include layout optimization and smart factory.

Namwoo Kang received his PhD in Design Science from the University of Michigan. He is currently an Assistant Professor of the Cho Chun Shik Graduate School of Green Transportation at Korea Advanced Institute of Science and Technology (KAIST). His research interests include generative design, data-driven design, artificial intelligence, and design optimization.

Dr. Kokkolaras is the founding director of the Systems Optimization Laboratory in the Department of Mechanical Engineering at McGill University. He is also Associate Director of the McGill Institute for Aerospace Engineering (MIAE) and a full member of the multi-university Group for Research in Decision Analysis (GERAD) funded by the Province of Quebec. His research interests include multidisciplinary optimization, simulation-based engineering design, uncertainty quantification, modeling and validation, systems of systems, product families and optimization applications in engineering. Dr. Kokkolaras has co-authored more than 130 papers in peer-reviewed journals and conference proceedings (as well as 5 book chapters) and has given more than 50 invited talks at academic, industry and government seminars and workshops. He is an ASME Fellow, an Associate Fellow of the AIAA (serving on its Multidisciplinary Design Optimization Technical Committee since 2006), and elected member of the Advisory Board of the Design Society.

Panos Y. Papalambros is the James B. Angell Distinguished University Professor, the Donald C. Graham Professor of Engineering, and Professor of Mechanical Engineering; Architecture; and Art and Design, at the University of Michigan. He conducts research in design science and systems optimization, with applications to automotive systems and sustainable products. He is a graduate of Stanford University and the National Technical University of Athens, and co-author of the textbook Principles of Optimal Design: Modelling and Computation (2017, 3rd ed.).

Abstract

Optimal system design at the conceptual functional level, i.e., before the embodiment of the functions is determined in detail, focuses primarily on performance. Embodiment determines the geometry and position of subsystems and components, which must be packaged usually within strict spatial envelops to achieve compactness or other external requirements such as styling. Packaging objectives and constraints may therefore compete with performance ones, leading to redesign and costly delays if these conflicts are not addressed early in the design process. This paper presents a design optimization framework for coupled performance and packaging problems. Using the cooling system for a heavy duty tracked series hybrid electric vehicle as an example, we demonstrate the framework combining commercial CAD software with optimization tools and including pipe routing which is a basic requirement in many mechanical systems.

Keywords: performance; packaging; pipe routing; layout optimisation: system design optimisation; hybrid electric vehicle; cooling system

1. Introduction

Increasing market demand on smaller, more compact products with the same or better performance has been making system design more challenging. To meet these market requirements, a typical mechanical system tied to both functionality (conceptual design) and geometric realization (embodiment design) is depicted in Fig. 1.

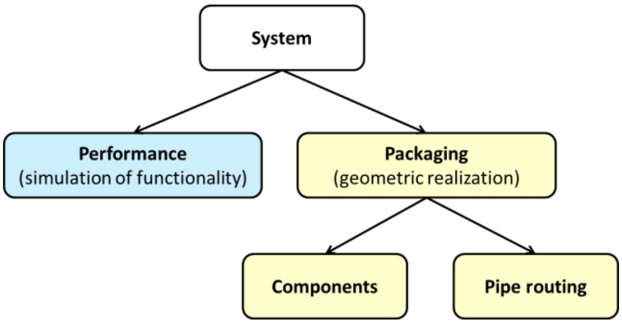


Fig. 1. System design considering performance and packaging

Designing the system to meet system performance requirements is a fundamental engineering task that may start during conceptual design but continues through detailed embodiment design of subsystems and components and their interactions. These interactions include geometric ones that are often critical in the ultimate success of the design.

Optimization models for system performance are well developed and used widely (Papalambros and Wilde 2017; Lewis et al, 2006). Multiple models are often needed for different types of analysis. Commercial software can potentially address model integration at the functional performance level. Accounting for the geometric interaction in the embodiment, such as interference among system components, is crucial because interference makes the design infeasible. Constraints are then added usually as simple upper and lower bounds or simple analytic equations for some geometric dimensions. This way of constraining decisions is rather crude and may not capture important system interactions, so that interference is discovered later in the product development process with increasing costs of redesign (Thomke and Fujimoto, 2000).

Packaging components is an optimization process that finds a desirable placement for the system's components within a given space such that a set of objectives is optimized, while satisfying spatial constraints. In the literature, packaging is also referred to as packing, layout design, configuration design, and spatial engineering. Packaging problems have been studied in many applications, such as electrical circuit layout, glass or metal cutting, truck loading (Grignon et al., 1996), trunk packing (Ding and Cagan, 2003; Eisenbrand et al., 2005; Tiwari et al., 2008), rapid prototyping (RP) (Aladahalli et al., 2003; Hur et al., 2000; Ikonen et al., 1996, 1997, 1998), architectural floorplan layout (Michalek, et al. 2002; Michalek and Papalambros, 2002), routing (Hills and Smith, 1997; Sandurkar et al., 1997; Szykman and Cagan, 1996; Szykman et al., 1998), and mechanical component layout.

Packaging has been an important issue in vehicle design optimization at the vehicle level and system level. Mastinu and Gobbi (2001) solved the layout optimization problem for railway passenger vehicles using Genetic Algorithm (GA) to minimize life cycle costs, with respect to vehicle layout such as vehicle length, number of bogies, seat length, etc. Segeborn et al (2007) proposed a dynamic packing method which generates collision-free assembly paths by automatically shrinking geometries, and applied the proposed method to the rear door assembly of a vehicle. Dong et al (2011) optimized the vehicle underhood layout by shape morphing through bi-level decomposition-based optimization. Wysocki et al. (2020) optimized the suspension layout to reduce interior road noise with automatically modifying FE model to fit new kinematics.

Packaging problems in mechanical system design are more challenging than 2D applications such as circuit layout or metal cutting, due to larger dimensionality and increased geometric complexity. Complex 3D geometry leads to increased computational time for interference checking, which is inevitable for finding a feasible layout. Detailed 3D CAD models, however, are not required or not available at the preliminary design stage. Therefore, abstract representation of the components is necessary during this early layout process. These abstract shapes are used to check for interferences between components during optimization and are later replaced with the original geometry for more accurate but time-consuming computation. To capture the design intent at the early design phase, these abstract models should balance accuracy of geometry representation with rapid computation.

Packaging problems deal generally with fixed shapes of components derived from component design. Although the shapes of components are changed during optimization (see, e.g., Yin et al., 2004; Dong et al., 2011), the shapes are not linked with performance. This lack of linking is due, to a large extent, to the difficulty in solving the packaging problem. As noted above, however, ignoring or not properly accounting for packaging can have detrimental effects on the downstream design activities. Therefore, coupled optimization of system performance and packaging is desirable.

Furthermore, pipe routing plays an important role in packaging problems because the additional space necessary for pipe location and assembly can be significant. In addition to avoiding overlapping, packaging must allow feasible paths for pipes among the components. Pipe paths (routes) are generally divided into two shapes, orthogonal and non-orthogonal. Orthogonal routes bend 90 degrees and are widely used in chemical plant design (Guirardello and Swaney, 2005). In mechanical system design, orthogonal routes are impractical, for example, Szykman and Cagan (1996) developed a pipe routing algorithm to create non-orthogonal routes. The pipe routing problem can be formulated as an optimization problem with the number of bends and bend locations as variables (e.g., Sandurkar, 1997; Szykman et al., 1998; Yin and Cagan, 2000), and non-gradient methods used for its solution. On the other hand, as pointed out in Zhu and Latombe (1991), pipe routing problems can be seen as robot path planning ones. When a set of obstacles, the robot initial position, and the robot final position are given, robot path planning algorithms find a path that moves the robot from the initial to final position avoiding the obstacles. When a spherical robot with three degree of freedom for translation in x, y, and z directions finds a path, the solution is the same as for pipe generation. In the present study we adopted the robot path planning approach to address the pipe routing problem.

The optimization framework in this paper consists of two steps: (a) performance and packaging optimization, (b) packaging and pipe routing optimization. We demonstrate the proposed framework and computation environment through application to the cooling system for a heavy duty tracked series hybrid electric vehicle (SHEV). A summary description of the SHEV system is as follows.

This study uses the cooling system architecture created based on the power management modes of the SHEV, as shown in Fig. 2, which is one of the architectures proposed by Park et al. (2007). The engine, generator, charge air cooler (CAC), and oil cooler are integrated in one cooling tower (Tower 1) and the power bus and motor are integrated into another cooling tower (Tower 2) in the architecture. Fig. 3 shows the 3D representation and geometry of the architecture.

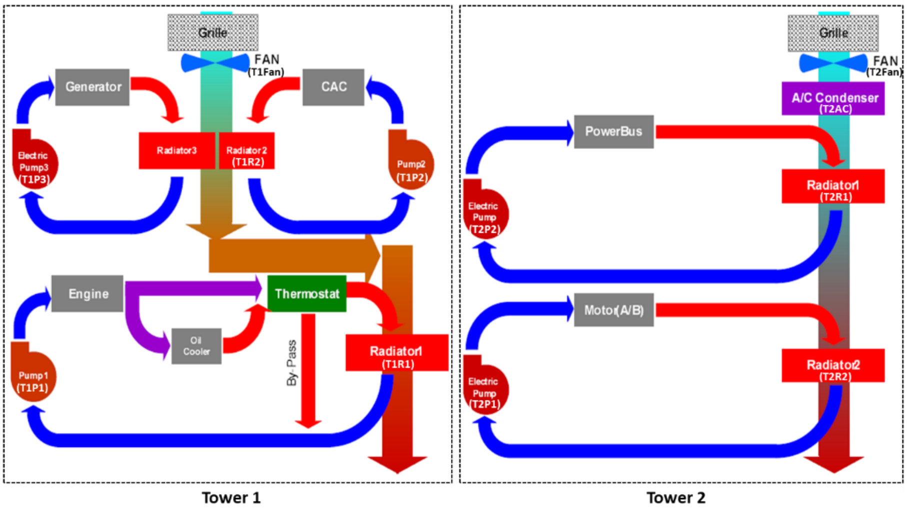
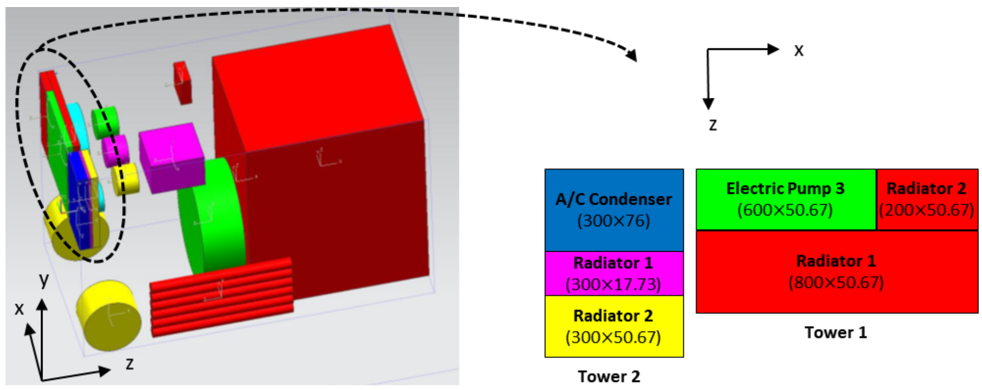


Fig. 2. Architecture of the cooling system of SHEV

Fig.3(a) is a 3D representation of the architecture, with each box and cylinder representing the major components that make up the cooling system. These components must not overlap with each other. For example, when the front part corresponding to the dotted line is displayed on the *xz*-plane, this can be represented in Fig. 3(b), showing the initial geometry of each component. More detailed geometry will be introduced in Section 2.2.



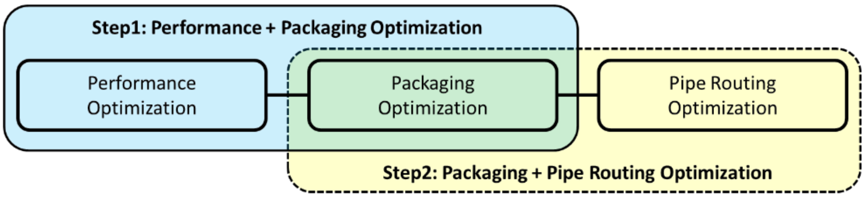
(a) 3D representation of the architecture (b) 2D geometry of the front part

Fig. 3. 3D representation and geometry of the architecture

The remainder of the paper is organized as follows. Section 2 introduces the proposed design optimization framework and problem formulations; Section 3 presents the simulation modeling and computational environment for the application study; Section 4 provides results and discussion; Section 5 concludes with a summary, contributions, and future work.

2. Methodology

2.1. Integrated Design Optimization Framework



Problem	Performance Optimization	Packaging Optimization	Pipe Routing Optimization
Objective	Minimize power consumption	Minimize distance among components	Minimize length of pipes
Variables	Sizes of components	Positions and orientations of the components	Pipe route
Constraints	Temperatures of components & non-overlapping	Non-overlapping	Non-overlapping

Fig. 4. Integrated design optimization framework for performance, packaging, and pipe routing

The design framework consists of two steps as shown in Fig. 4, with each step comprising a multi-objective problem: performance and packaging in Step 1, and packaging and pipe routing in Step 2. Packaging optimization is included in both steps, so that it is optimized twice in sequence. This choice is motivated by the application study, the cooling system for a heavy duty tracked series hybrid electric vehicle (SHEV) and can be applied to designs of other vehicles. For such problems, it is often more efficient to divide the optimization process into two steps due to problem complexity and compatibility of the computational environments of commercial CAD/CAE software. If problem complexity and software availability would allow simultaneous optimization of all three problems, this would be the preferred strategy. Here we focus on developing a practical strategy that utilizes existing commercial software but offer no claim of proving that the strategy locates the global system optimum.

Performance optimization determines the sizes of components such as pumps, fan, and radiators for minimizing the overall power consumption, subject to non-overlapping constraints and the target temperatures for components. Packaging optimization determines positions and orientations of the components for minimizing distance among components, subject to non-overlapping constraints. The proposed strategy can handle almost all geometric requirements at both component and assembly level, if experts can formulate the appropriate constraints and objective functions. The pipe routing optimization finds the optimal pipe route for minimizing length of pipes subject to non-overlapping constraints. As noted, the pipe routing problem is solved by implementing a robot motion planning algorithm.

For performance optimization in the application study, we adopted the cooling system simulation model developed by Park and Jung (Park et al., 2007; Park and Jung, 2008; Park and Jung, 2010). That model did not consider geometry of components, and so we integrate this model with the packaging simulation to optimize both performance and packaging. The cooling system simulation requires expressing the component operating conditions as a function of time to simulate the thermal response of the cooling system when the vehicle is driven over a driving cycle. The cooling system has many components: coolant pumps, fans, radiators, thermostats, and heat sources. Each component is modeled to predict its thermal response and/or power consumption. In Fig. 2, the optimized components indicate their design variable symbols in parentheses, with the symbol descriptions given in Table 1. A detailed description of the simulation model is given in Section 3.

2.2. Step1: Performance + Packaging Optimization

The multi-objective problem with performance and packaging objectives, f_p and f_g , respectively, is stated as follows.

$$\min_{\mathbf{x}=[\mathbf{x}_p,\mathbf{x}_g]} f_p(\mathbf{x}) + f_g(\mathbf{x})$$

subject to

$$g_1 = \sum_{i=0}^{N-1} \sum_{j=i+1}^{N} Vol(C_i \cap C_j) \leq 0$$

$$g_2 = T_{gen} - T_{gen\_target} \leq 0$$

$$g_3 = T_{eng} - T_{eng\_target} \leq 0$$

$$g_4 = T_{CAC} - T_{CAC\_target} \leq 0$$

$$g_5 = T_{oil} - T_{oil\_target} \leq 0$$

$$g_6 = T_{pb} - T_{pb\_target} \leq 0$$

$$g_7 = T_{mot} - T_{mot\_target} \leq 0$$

where (1)

$$f_p(\mathbf{x}) = P_{T1} + P_{T2}$$

$$P_{T1} = P_{T1P1} + P_{T1P2} + P_{T1P3} + P_{T1Fan}$$

$$P_{T2} = P_{T2P1} + P_{T2P2} + P_{T2Fan}$$

$$f_g(\mathbf{x}) = d_1 + d_2 + 10 \times d_3 + d_4 + d_5 + d_6$$

$$d_1 = \|\mathbf{P}_{gen} - \mathbf{P}_{T1R3}\| + \|\mathbf{P}_{T1R3} - \mathbf{P}_{T1P3}\| + \|\mathbf{P}_{T1P3} - \mathbf{P}_{gen}\|$$

$$d_2 = \|\mathbf{P}_{CAC} - \mathbf{P}_{T1R2}\| + \|\mathbf{P}_{T1R2} - \mathbf{P}_{eng}\| + \|\mathbf{P}_{eng} - \mathbf{P}_{CAC}\|$$

$$d_3 = \|\mathbf{P}_{gen} - \mathbf{P}_{T1R1}\|$$

$$d_4 = \|\mathbf{P}_{pb} - \mathbf{P}_{T2R1}\| + \|\mathbf{P}_{T2R1} - \mathbf{P}_{T2P1}\| + \|\mathbf{P}_{T2P1} - \mathbf{P}_{pb}\|$$

$$d_5 = \|\mathbf{P}_{motA} - \mathbf{P}_{motB}\| + \|\mathbf{P}_{motB} - \mathbf{P}_{T2R2}\| + \|\mathbf{P}_{T2R2} - \mathbf{P}_{T2P2}\| + \|\mathbf{P}_{T2P2} - \mathbf{P}_{motA}\|$$

$$d_6 = \|\mathbf{P}_{eng} - \mathbf{P}_{oil}\|.$$

T represents the temperature of a component. P represents power consumption. d indicates the distance between positions $\mathbf{P}$ of respective components, where $\|\cdot\|$ denotes L^2-norm. *T1*, *T2*, *P1*, *P2*, *R1*, and *R2* represent Tower 1, Tower 2, Pump 1, Pump 2, Radiator 1, and Radiator 2, respectively. *gen*, *eng*, *CAC*, *oil*, *pb*, *motA*, and *motB* represent generator, engine, charge air cooler, oil cooler, power bus, motor A, and motor B, respectively. Architecture of these components are shown in Fig. 2.

The objective function for performance, f_p, is to minimize the power consumed by the fans and pumps in the cooling system. The power consumption is computed by the vehicle simulation under the grade load condition which is the most severe condition for the cooling system. The total power consumption is calculated as the sum of the power consumption of the two towers, T1 and T2.

The design variables for performance, $\mathbf{x}_p$, are the sizes of the radiators and pumps. Changes of 10% from the baseline design are used in our study arbitrarily. These percentages should be decided by designers based on historical design data or domain knowledge, and these can be affected by design conditions and capabilities. The size of the pump is not the actual geometry size but the scaling factor of the pump's capacity. The design variables that appear in Eq. (1) are described in Table 1 along with their lower and upper bounds; TiAC is the charge air cooler in tower i, TiFan is the fan in tower i, TiPj the pump j in tower i, and TiRj the radiator j in tower i. The first eight design variables are for Tower 1, and the rest are for Tower 2.

Table 1. Design variables for optimization problem for performance

Variable	Lower bound	Upper bound	Description
$x_1 = S_{\mathrm{T1P1}}$	0.9	1.1	Scaling factor for T1P1
$x_2 = S_{\mathrm{T1P2}}$	0.9	1.1	Scaling factor for T1P2
$x_3 = S_{\mathrm{T1P3}}$	0.9	1.1	Scaling factor for T1P3
$x_4 = S_{\mathrm{T1R1X}}$	0.9	1.1	Scaling factor for width of T1R2
$x_5 = S_{\mathrm{T1R1Z}}$	0.9	1.1	Scaling factor for thickness of T1R1
$x_6 = S_{\mathrm{T1R2X}}$	0.9	1.1	Scaling factor for width of T1R2
$x_7 = S_{\mathrm{T1R2Z}}$	0.9	1.1	Scaling factor for thickness of T1R2
$x_8 = S_{\mathrm{T1Fan}}$	0.9	1.1	Scaling factor for T1Fan
$x_9 = S_{\mathrm{T2P1}}$	0.9	1.1	Scaling factor for T2P1
$x_{10} = S_{\mathrm{T2P2}}$	0.9	1.1	Scaling factor for T2P2
$x_{11} = S_{\mathrm{T2R1X}}$	0.9	1.1	Scaling factor for width of T2R1
$x_{12} = S_{\mathrm{T2ACZ}}$	0.9	1.1	Scaling factor for thickness of T2AC
$x_{13} = S_{\mathrm{T2R1Z}}$	0.9	1.1	Scaling factor for thickness of T2R1
$x_{14} = S_{\mathrm{T2R2Z}}$	0.9	1.1	Scaling factor for thickness of T2R2
$x_{15} = S_{\mathrm{T2Fan}}$	0.9	1.1	Scaling factor for T2Fan

The first constraint, g_1, is a non-overlapping constraint. It is in a general form for generic shapes and locations; C indicates each component, and the constraint means that the overlapping volume of different components must be less than zero. This constraint affects both performance and packaging objective functions. The temperatures of the components that should be maintained by the cooling system are the constraints. The target temperatures T for the components in g_2 to g_7 are listed in Table 2. The temperature distribution in a heat source including the engine, generator, motors and power bus should be minimized by the cooling system because large temperature distribution can deteriorate the durability of the heat source component.

Table 2. Control target temperatures of the components

Component	Control target temperature (℃)
Engine	120
Motor	95
Generator	95
Oil cooler	125
Power bus	70
Battery	45

The objective function for packaging, f_g, minimizes the summation of distances among components, d, in the same cooling circuits including the oil cooler. Since the distance between engine and Radiator 1 in Tower 1 ($T1R1$) plays an important role for system compactness, the weight 10 is applied to d_3. **P** represents the position of the component, where $\|\cdot\|$ denotes L^2-norm.

The design variables for packaging, $\mathbf{x}_g$, are the positions and orientations of the components. Some parts such as radiators and fan can be grouped together. The number of design variables are automatically calculated from the DOFs in the input file. The case study has 17 design variables, as listed in Table 3. The shape of the components are parameters for the packaging problem. Given a position and orientation, a 4 by 4 transformation matrix is built and set to the assembly, not each component in the CAD system.

Table 3. Design variables for packaging optimization

No.	Name	Initial size (W×H×L* or D×L** (mm))	Degree of freedom
1	Container	2000 × 1500 × 2500	-
2	Engine	1270 × 1016 × 1346	t_x
3	Generator	800 × 300	(relative to engine)
4	CAC	200 × 200 × 76	t_x, t_y, t_z
5	Oil cooler	75 × 1000	t_x
6	T1R1	800 × 600 × 50.67	-
7	T1R2	200 × 600 × 50.67	(relative to T1R1)
8	T1R3	600 × 600 × 50.67	(relative to T1R1)
9	T1Fan	500 × 100	(relative to T1R1)
10	T1P3	200 × 150	t_x, t_y, t_z
11	Power bus	365 × 238 × 380	t_x, t_y, t_z
12	Motor (A/B)	405 × 241	-
13	T2R1	300 × 600 × 17.73	-
14	T2R2	300 × 600 × 50.67	(relative to T2R1)
15	T2AC	300 × 600 × 76	(relative to T2R1)
16	T2Fan	200 × 50	(relative to T2R1)
17	T2P1	200 × 150	t_x, t_y, t_z
18	T2P2	200 × 150	t_x, t_y, t_z

* for box shape components
** for cylinder shape components

2.3. Step2: Packaging + Pipe Routing Optimization

For Step 2, packaging optimization is performed again with pipe routing optimization. The size of components is given from the optimization results of Step 1. Although pipe performance is not explicitly included in the model, it is assumed that the minimum length design is preferred. Pressure drop models for pipes can be added to the performance evaluation at a later stage based on the pipe length created in the geometry problem. Pipe generation is only executed when there is no interference. The optimization problem is formulated as:

$$\min_{\mathbf{x}_g} f_g(\mathbf{x}_g) + L_{pipe}$$

subject to

$$g_1 = \sum_{i=0}^{N-1} \sum_{j=i+1}^{N} Vol(C_i \cap C_j) \leq 0$$

where (2)

$$L_{pipe} = \begin{cases} \infty, & if\ g_1 \neq 0 \\ \sum_{i=1}^{N_{pipe}} L_i & if\ g_1 = 0. \end{cases}$$

The summation of all pipes' length L_{pipe} is added to the objective function of the packaging optimization problem, f_g, to minimize the total pipe length. L_{pipe} goes to infinity if the non-overlapping constraint, g_1, is not satisfied so that pipes are not generated. The design variables, $\mathbf{x}_g$, are the same as those of the packaging problem in Step 1. Given the location input and output ports, the pipes are automatically generated while avoiding collision with the components and the previously generated pipes, and so design variables such as the number and location of bends are not included. Also, the locations of engine and oil cooler are fixed at the optimized values of the packaging problem. In the study, several connections from the total of 15 pipe routings are excluded from the optimization model because either their pipe diameters are smaller than the others or the locations of the two connected components are fixed. Seven pipe routings are generated during optimization.

The predetermined order of pipe generation based on these heuristics is placed into the optimization problem. Once the location of components is fixed, the pipes are automatically generated by the given order. Thus, the initial pipe path is determined by the initial positions of components. If the specific location must be on the path for a certain pipe as an additional constraint, the location can be inserted between input and output ports so that the two pipes are internally generated. This particular functionality was not implemented in the results present here but it is a straightforward future enhancement.

Pipes are generate using the sampling-based algorithm which searches for collision-free paths only by sampling points (Choset et al., 2005). In this case, the generated piping shapes can be different for each optimization run. However, the shapes generated show that there is at least a feasible solution for placing the pipes in the system embodiment given the locations of its components. In addition, since the method of connecting collision-free points alone cannot make the path smooth, this study applies a smoothing function to refine the path (MPK, 2019).

3. Modeling

This section discusses the computational implementation for the application study. We replace part of the previous models (Park et al., 2007; Park and Jung, 2008, 2010) with a surrogate model to reduce computational effort and allow simultaneous performance and packaging optimization. Pipe routine is generated from a given CAD model using a robot path planning algorithm. Lee (2014) includes more modeling details.

3.1. Performance Simulation and Surrogate Modeling

3.1.1. Vehicle Simulation Setting

Based on the configuration of vehicle components and the power management modes, a vehicle model with a SHEV propulsion system is configured employing the vehicle-engine simulation (VESIM) previously developed at the Automotive Research Center (ARC) at the University of Michigan (Park et al., 2007). The VESIM model is used to acquire the operating conditions of SHEV powertrain components. The resulting component operating conditions are provided as input data to the cooling system simulation.

The specifications of the SHEV simulated in the example are summarized in Table 4. A turbocharged diesel engine is chosen as the power source due to its better efficiency vs. a spark ignition engine and its lower cost vs a gas turbine (Park and Jung, 2008). The rated engine power is determined based on the power (kW) to weight (ton) ratio of 15. Generator and motor capacities are determined to convert all the power supplied by the engine to electricity. Two alternating current (AC) induction type electric motors are used to drive two separate tracks of the vehicle and a lead-acid battery is selected.

Table 4. Specification of the selected SHEV

Component	Type	Specification
Vehicle	Tracked SHEV	20 ton
Engine	Turbocharged diesel	300 kW
Generator	Permanent magnetic	300 kW
Motor	AC induction	2 × 150 kW
Battery	Lead-acid	18 Ah/120 modules

The capacity of a cooling system should be enough to remove all of the heat generated by the heat sources under extreme operating conditions. Since the grade load condition is the most severe condition for the cooling system, this condition is used for design and evaluation, and the vehicle speed, road profile, and ambient temperature are set to 48 km/h, 7% (uphill), and 40℃, respectively. In the grade load condition, the simulation results show a repetition pattern, so 10 minutes is used to evaluate the cooling system in order to reduce computation time.

The cooling simulation model was built in Simulink (Mathworks, 2019) and represents the relationships among the components in terms of thermal performance. The power consumption of the cooling system, which is the objective function of performance optimization in Eq. (1), is computed by this Simulink simulation model. However, the optimization model for packaging is necessary in order to check if there is a feasible layout solution and further optimize it.

3.1.2. Surrogate Modeling

Radiator simulation models from previously published research is not suitable for an optimization study because of long simulation times. The original model was developed using Finite Difference Method (FDM) with staggered grid system (Jung and Assanis, 2006) and has a high computational cost relative to the other analysis models in our present modeling framework. In addition, this FDM model is used in 5 radiators and a charge air cooler (CAC) with different input parameters, which makes computational cost more expensive. For this reason, we developed surrogate models for the radiator performance.

The FDM model has 23 inputs and 5 outputs, where the input parameters include the dimensions for core, tube, and fins, temperature and mass flow rate of coolant, and velocity and temperature of air. Among 23 inputs, 7 variables are selected to generate the surrogate model because other values are either constant or calculated based on the given values. Among the input parameters, the range of some values such as temperature and mass flow rate of coolant cannot be estimated without running the simulation model because these values are internally calculated during the vehicle simulation. Vehicle simulations are performed with the three different input values over the 30 minutes driving cycle, which is the baseline design, and $\pm$10% from the baseline design. The ranges of input variables are chosen based on the simulation results. The input and output parameters are summarized in Table 5. Note that not all seven input variables are used in all models. For example, the core size of CAC is constant in the simulation, so those variables are excluded.

Table 5. Input and output variables for DOE study

Inputs	Outputs
core width	
core height	heat dissipation (q_{rad})
core thickness	outlet air temperature (T_{air})
coolant mass flow	outlet coolant temperature(T_{cool})
inlet coolant temperature	air pressure drop (ΔP_{air})
inlet air temperature	coolant pressure drop (ΔP_{rad})
inlet air velocity	

8,000 design points were generated using a Latin hypercube design of experiments (DoE); any points that caused numerical errors were excluded. A surrogate model was built for each radiator model including the CAC model. Two thirds of the points were used for training and the remaining third was used for testing (cross-validation).

We compared the performance of a neural network model and a polynomial one. The neural network had two hidden layers and the polynomial function had second order and interaction terms for each radiator. The polynomial model was selected as the surrogate because it showed better performance in terms of R-squared values and computational time. The R-square values are shown in Table 6.

Table 6. R-squared values for polynomial model

Component	q_{rad}	T_{air}	T_{cool}	ΔP_{air}	ΔP_{rad}
T1R1	0.97	0.94	0.83	0.89	0.96
T1R2	0.98	0.84	0.98	0.92	0.99
T1R3	0.92	0.81	0.95	0.91	0.99
T1CAC	0.99	0.99	0.94	0.99	0.99
T2R1	0.98	0.99	0.96	0.74	0.99
T2R2	0.98	0.99	0.96	0.74	0.99

3.2. Packaging Simulation and Computational Environment

3.2.1 Component Geometry Setting

Component shape and location are not available in the Simulink model, and so the abstract models must be assumed based on existing models, experience, and market availability. In our study, initial shapes and sizes of components are assumed based on the reference model used in the performance model or on similar components from catalogs as much as possible.

The computational environment must incorporate changes of geometry and size. The most important size is the engine room which is selected based on vehicle specifications. The shapes are assumed as boxes. The look-up table for the engine is generated for a heavy duty, inline, six cylinder, turbocharged, and intercooled diesel engine with data from engine dynamometer tests (Park and Jung, 2010). As shown in Park et al. (2007), turbocharged diesel engines are widely used in similar tracked vehicles, and the specifications of the baseline diesel engine described in Park and Jung (2010) are used to determine the engine to be used, namely, the Detroit Diesel DD13 (Detroit DD13 Engine, 2019) shown in Fig. 5. The initial sizes used in the model are summarized in Table 3.

Specifications	
Configuration	Inline 6 Cylinder
Displacement	12.8L
Compression Ratio	17.3:1
Bore	132mm
Stroke	156mm
Weight (Dry)	1152kg
Electronics	DDEC VI
Clutch Engagement Torque	950 lb-ft
Oil Capacity	39.5L
Dimensions (H x W x L)	53.0 x 51.5 x 58.3 in.

Fig. 5. Engine specifications (Detroit DD13 Engine, 2019)

3.2.2 Computational Environment

The shapes and locations of the components should be modeled in commercial CAD software with dimensions so that the size changes in the simulation model are easily applied to the shapes in the packaging problem. The process and assumptions of the computational environment are as follows.

- *Create geometries:* The component shapes are assumed and represented by primitives such as box, cylinder, and sphere or by abstract geometry such as bounding box and convex hull. Furthermore, when the bounding box or convex hull is calculated, these geometries are created in the system.
- *Modify geometries:* The component shape is changing during the optimization. This function can be achieved by modifying dimensions, if dimensions exist in the model.
- *Load/Import existing CAD files:* Components can be designed in heterogeneous systems, so they inevitably exist in various file formats. Neutral formats such as IGES (Initial Graphics Exchange Specification) or STEP (Standard for the Exchange of Product model data) can be used, so the function of importing a neutral file format is required.
- *Transform components:* The 4 x 4 transformation matrix containing translation and rotation information is applied to the components, which can be a single solid body or an assembly.
- *Handle assembly structures:* An assembly is treated as one component during the packaging. For example, a radiator assembly consisting of radiator, fan, and shroud.
- *Calculate mass properties:* Mass properties such as volume, moment of inertia, and center of mass are important quantities in the packaging problems. Calculation of these quantities are available based on the type of geometric representation.
- *Check interference:* This is a key function required in the computational environment.
- *Visualize geometry:* Visualizing and examining the packaging result on the screen are necessary.

To satisfy the above requirements and to develop a practical, compatible system, we customized and integrated the existing commercial CAD software, programing, and tools. NX V6 (NX, 2019) and Microsoft Visual Studio 2005 were chosen as CAD modeler and development tool, respectively. Another advantage of using NX V6 API is that the software can be run in batch mode, an important feature for use with commercial software integrators such as MATLAB, iSIGHT (2019), and OPTIMUS (2019).

Three key computational environments were developed.

First, the object-oriented wrapper class using C++ was developed for ease of project development and depicted in Fig. 6. Direct use of the NX API functions in the packaging optimization problem is possible. However, the wrapper class provides higher level capabilities and easier development. Moreover, it allows focus on specific problems by the user instead of understanding the CAD system. Individuals with less knowledge of the CAD system and API can still perform their research in virtue of this class.

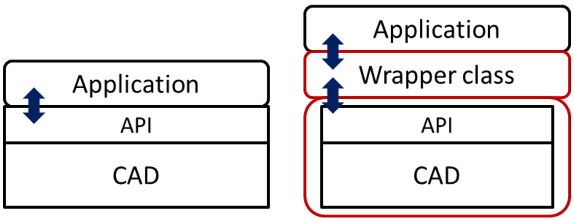


Fig. 6. The wrapper class architecture

Second, we integrated the optimization algorithms within C++. Considering popular graphic library and API supported by CAD modeler, C++ was chosen to implement the computational environment, which means optimization codes written in C or C++ were required. Two algorithms were integrated in the system: CFSQP, a C code for Feasible Sequential Quadratic Programming (Lawrence et al., 1997), and Genetic Algorithm (GA) (KanGAL, 2019).

Third, a MATLAB interface was developed. Since a commercial CAD modeler using C/C++ API was selected as the computational environment, methods to interface with the MATLAB and Simulink were needed. Figure 7 shows the framework developed to call MATLAB functions from the environment using NX API. The application writes the input file that contains the design variables for the optimization problem and executes RunML.exe that is the standalone executable program that reads the input text file, then calls MATLAB functions via COM to calculate objective and constraint functions.

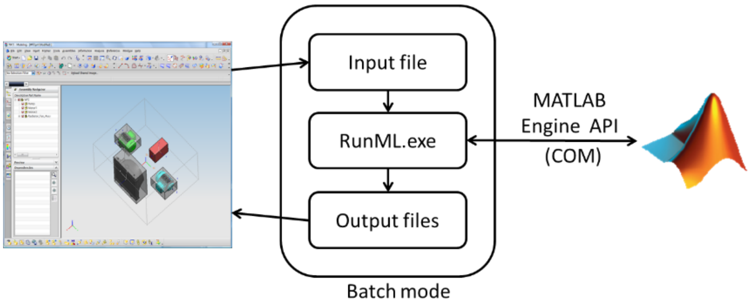


Fig. 7. Proposed framework for MATLAB interface

3.3. Pipe Routing and Computation Environment

3.3.1. Pipe Routing Algorithm

Pipe routing finds the shortest path avoiding interference with components and other pipes and cables. As noted earlier, this problem has been widely studied in many fields such as cable routing in electrical engineering, chemical and power plant design, building design, and automobile and aerospace industry. It is an NP-hard problem whose solution heavily depends on human experience (Yin et al., 2010).

Pipes, hoses, and electrical cables in a system performance model are generally represented by mathematical lines with no volume and mass. In reality, a packaging problem in a mechanical system cannot be completed because the space necessary for pipe placement and assembly is not negligible. Feasible paths for pipe are necessary in order to have a feasible layout for the system.

Fast generation of feasible pipe routes is needed in order to integrate the pipe routing problem into the optimization framework and determine a good estimate of pipe length. Therefore, pipe route generation is formulated as an analysis function not as an optimization problem, as discussed in Section 2.3.

The order of pipe generation is important because the pipes previously generated are used as obstacles for the next pipe generation. Human experience, such as heuristics of working "from inside to outside", "from thick pipe to thin pipe," and "from short pipe to long pipe," are used to determine the piping order (Zhou and Yin, 2010). In this paper, the piping order is predetermined and given to the problem as a parameter.

The approximate evaluation of pipe length and pipe locations at the early stage of design is sufficient in this application study. Robot motion planning has been the subject of much research including sampling-based algorithms (e.g., Choset et al., 2005) that have been most successful for high degree-of-freedom motion planning problems. In the present study we adopted and integrated an existing algorithm as described below.

3.3.2. Computational Environment

We used the Motion Planning Kit (MPK) which is a C++ library and toolkit for developing single and multi-robot motion planning applications (MPK, 2019). In addition to MPK, two more separate libraries are required to link MPK, Coin3D/SoWin (Kongsberg Oil & Gas Technologies, 2010) and Proximity Query Package (PQP) (Gottschalk et al., 1999). MPK uses Coin3D/SoWin for GUI and graphic library, and POP (POP Collision Detect, 2019) for collision detection.

The library for a robot motion planning algorithm was then integrated into our computational environment and optimization problem for pipe routing as follows.

First, the port names and diameter for a pipe are defined a priori, but the location of the ports will be evaluated during optimization. The reference points are generated in a CAD model and named. When the component is moved or resized, the reference point may be relocated because it is created based on the relations with the existing geometry entities such as vertices, edges, and faces. Once the reference points are created on the CAD files, the information for the pipe generation is added to the input file for optimization.

Second, since the NX CAD modeler does not support export to the Open Inventor file, we studied the Open Inventor file format and developed code to export the geometry to Open Inventor. Then, MPK generates the list of points for the pipe path, from which the pipe length is calculated. When MPK produces the list of points, the geometry of the pipe is created in CAD system and exported to MPK. Multiple solid bodies of cylinder shapes are created in a part file. After MPK library is successfully integrated and tested, the entire model code is called during optimization. In addition, we developed a batch mode program to link all models with MATLAB, so that they can be called from the main computational environment. Fig. 8 shows the interface between NX and MPK for pipe geometry generation.

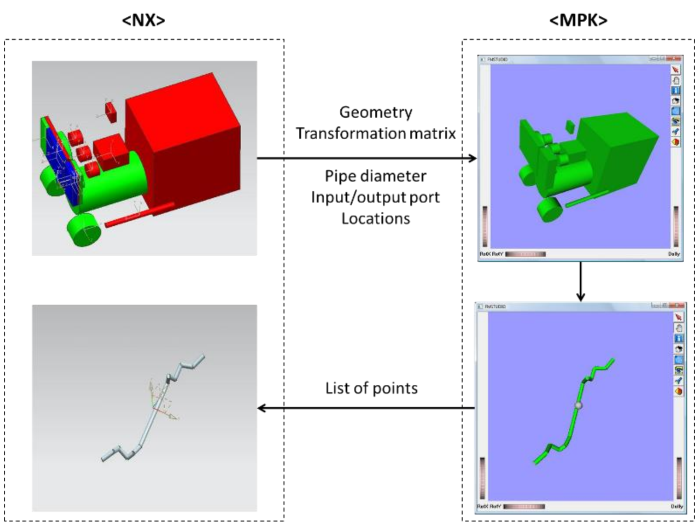


Fig. 8. Interface between NX and MPK for pipe geometry generation

4. Optimization Results and Discussion

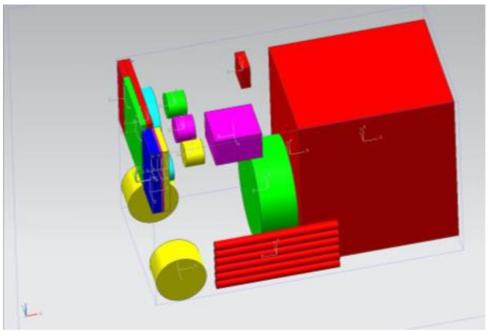
We now present and discuss the optimization results for each step obtained through the proposed framework.

4.1. Step1: Performance + Packaging Optimization

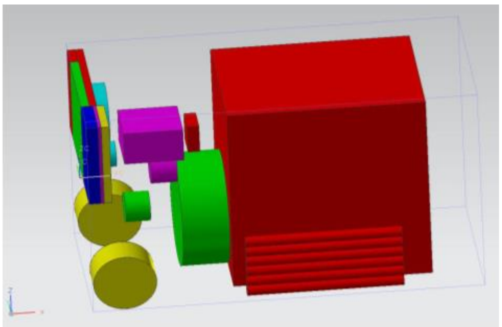
There are various strategies to solve the coupled optimization problem such as sequential, iterative, simultaneous, nested, and partitioned strategies (Peters, 2010). The sequential strategy is applied to this study due to the complexity of problem. In the sequential strategy each design is optimized once. The solution obtained from the previous problem becomes parameters for the following problem. In the study, the optimal sizes from performance optimization is applied to the shapes of components in the packaging problem as parameters. As noted earlier, this strategy is not guaranteed to determine the global system optimum.

Both a gradient-based algorithm (CFSQP) (Lawrence et al., 1997) and a derivative-free heuristic (a genetic algorithm) (KanGAL, 2019) were used to solve the problem (note that the non-overlapping constraint function is not smooth). For the GA, the population size was 170 and stopped at the 49th generation. The termination criterion was the relative percentage change in the objective function value being less than 0.001.

The initial layout is shown in Fig. 9(a). Some components such as fans, radiators, and engine are located based on knowledge of the system and assumptions of the performance problem. Other components such as electrical pumps and CAC are randomly placed while non-overlapping with other components. The optimization results are shown in Fig. 9(b). Through optimization, power consumption is improved by 25% while satisfying the temperature constraints, see Table 7. The optimal values of five design variables, x_1, x_2, x_8, x_9, x_{15}, are hitting lower and upper bounds. This indicates that the design space need to be revisited, and a parametric study for active constraints can be conducted. The goal of this paper is to propose a process which can identify components mostly affecting performance and packaging optimization. Compactness is improved by 10% while satisfying the non-overlapping constraint. We tested different initial values for optimization and present the best results among local optima.



(a) Initial design



(b) Step1: performance + packaging optimization

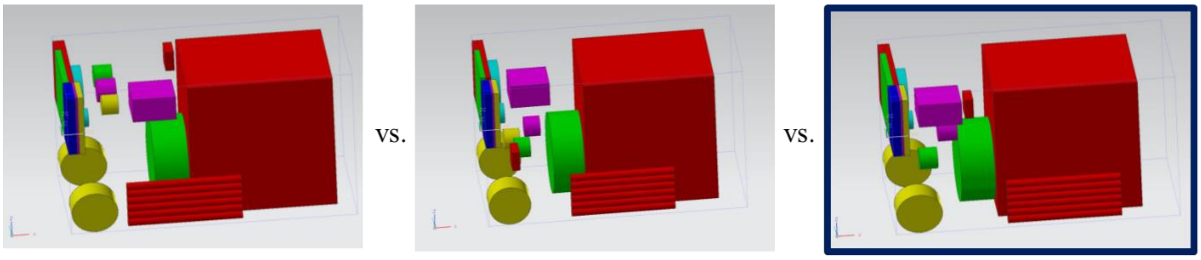
Fig. 9. Initial and the optimal packaging results of Step 1

Table 7. Optimal design values for performance optimization

Variable	Initial design	Optimal design
$x_1 = S_{\mathrm{T1P1}}$	1.0	0.90
$x_2 = S_{\mathrm{T1P2}}$	1.0	0.90
$x_3 = S_{\mathrm{T1P3}}$	1.0	1.05
$x_4 = S_{\mathrm{T1R1X}}$	1.0	1.08
$x_5 = S_{\mathrm{T1R1Z}}$	1.0	0.91
$x_6 = S_{\mathrm{T1R2X}}$	1.0	0.92
$x_7 = S_{\mathrm{T1R2Z}}$	1.0	1.08
$x_8 = S_{\mathrm{T1Fan}}$	1.0	0.90
$x_9 = S_{\mathrm{T2P1}}$	1.0	1.10
$x_{10} = S_{\mathrm{T2P2}}$	1.0	0.95

$x_{11} = S_{\mathrm{T2R1X}}$	1.0	1.09
$x_{12} = S_{\mathrm{T2ACZ}}$	1.0	1.00
$x_{13} = S_{\mathrm{T2R1Z}}$	1.0	1.09
$x_{14} = S_{\mathrm{T2R2Z}}$	1.0	0.98
$x_{15} = S_{\mathrm{T2Fan}}$	1.0	0.90

To gain further insights, we compare the results of the coupled performance and packaging optimization single objective optimization. Fig. 10(a) shows the result of performance optimization only, Fig. 10 (b) is the result of packaging optimization only, and Fig. 10(c) is the result of coupled performance and packaging optimization. In (c), the performance result is the same as in (a) since the performance and packaging problems are solved sequentially. However, the packaging results in (b) and (c) show a big difference. In (b), the component size is fixed because performance is not considered for packaging optimization. In (c), packaging is performed after the component size is changed through performance optimization. These results show the value of the coupled problem approach.

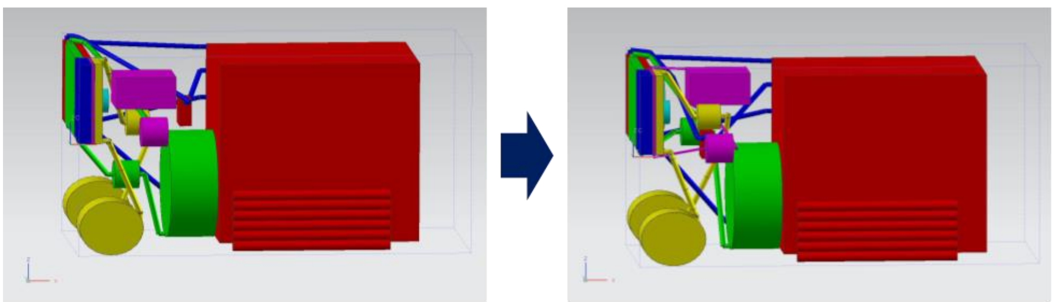


(a) Performance Only Optimization (b) Packaging Only Optimization (c) Performance + Packaging Optimization

Fig. 10. Comparison of optimization results

4.2. Step2: Packaging + Pipe Routing Optimization

In Fig. 11, (a) shows the initial design and (b) shows the optimal design of Step 2. The layout in (a) connects the pipeline to the optimized result of Step 1 without optimization. In (b), we performed packaging optimization and pipe routing again based on the optimized component size of Step 1 for improving compactness.



(a) Initial design (b) Step2: packaging + pipe routing optimization

Fig. 11. Initial and the optimal packaging results of Step 2

MPK is a sampling-based algorithm, so the generated path is different at each run and is created only when the layout has no overlapping. This revised optimization problem for packaging with pipe routing is solved with GA because the lengths of the pipes are discontinuous and non-smooth. The population size was 150 and GA stopped at the 20th generation. The termination criterion was that the relative change in percentage with respect to the previous best objective function value is less than 0.1.

At termination, compactness was decreased by 2% but pipe length was decreased by 38% while satisfying the non-overlapping constraint. Note that the algorithm terminated at the maximum number of generations allowed and as all GAs has no convergence criterion. The solutions are the best found rather than optimal in a mathematical sense. All 15 pipe routings were generated after the optimization to see whether the excluded pipes can be created in the system. Fig. 11(b) shows the final 15 pipe routing results.

4.3. Discussion

In Step 1, optimization of both performance and packaging can be performed within the same computation environment. Since a commercial CAD modeler and its API are used, most geometric requirements can be programmed and computed. By providing or being able to develop interfaces, integrations with other analysis tools are possible. Also, feedback from the result can easily be given to engineers because the software that they use for designing and optimizing products are the same. Note that implementing both problems requires knowledge of both domains, which means engineers should understand the performance model and know geometry and CAD modeler programming.

The optimization results also show that the coupled problem yields a better system solution than single objective optimization. However, the integrated optimization problem increases the number of design variables. If we deal with 20 components with 6 DOFs each, a total of 120 variables are added to a performance-only problem. In the study here, simultaneous optimization did not converge and the sequential strategy was the practical alternative, albeit without ascertaining optimality.

In Step 2, the optimization problems for both component packaging and pipe routing were solved within the same computation environment. The implementation issues as described in Section 3.3.2 were successfully addressed.

The optimization model found a better solution in terms of compactness than the result of Step 1 and demonstrated the usefulness of the proposed method and formulation. However, since a sample-based algorithm was used, maintaining a good path from the previous generation is not easy during iterations. To address this, each position of each bend needs to be stored in addition to pipe locations and orientations. This is not implemented here and would be a good future enhancement.

5. Conclusion

We presented a framework that couples packaging geometric considerations with performance optimization for a particular application, but the approach can be considered as sufficiently general. The combined optimization model is much more complex than the original performance model because it includes the additional objective function such as compactness, design variables such as positions and orientations, and constraints including a non-overlapping constraint. Also, other components in a system that are not used in the optimization model might need to be geometrically modeled to compute interference with existing components even though their sizes do not change.

Pipe routing problems are hard to solve but the generation of pipe and cable routes must be included in packaging of mechanical and electromechanical systems. Here we showed how we can estimate path feasibility and length of pipes for a given layout and include them in the packaging problem.

The practical implementation in a computational environment of separate computational tools, including computational geometry libraries, computer graphics (CG), and optimization algorithms presents challenges that are often very time consuming

The main contributions of this paper are summarized as follows.

First, the paper integrated a system's abstract representation suitable for simulation with the system's actual embodiment in its geometric physical instantiation; and used such integration for overall performance and geometric layout optimization. Second, a computational environment on a commercial CAD system was developed, which enables the optimization study that needs geometric computation with CAD models. Optimization algorithms were integrated in the computational environment: a gradient based algorithm, CFSQP, and a gradient-free algorithm, GA. For more convenient use of those algorithms and easy integration, the wrapper C++ classes were developed. Also, because a commercial CAD system was used as the foundation for the computational environment, the developed codes can be integrated more easily with the design process within a company. Third, pipe generation and pipe length estimates were addressed via integrating the motion planning algorithm. The pipe shapes were included in the packaging problem because the space required is not negligible. The library of motion planning algorithm was successfully integrated within the developed computational environment. Lastly, the developed framework was demonstrated on the cooling system for a heavy duty tracked SHEV.

The following issues require further investigation.

First, further research may examine the tradeoff between the number of convex geometries and computation time for both interference check and optimization. Second, true optimality of the system might be addressed using decomposition-based optimization strategies that have known convergence properties such as Analytical Target Cascading (ATC) (e.g., Kim, 2001; Kang et al., 2014). Lastly, a tighter integration of the path planning algorithm possible without adding another stand-alone problem as implemented in this paper, will reduce the compilation and computational effort.

Reference

Chandankumar Aladahalli, Jonathan Cagan, and Kenji Shimada, "Minimum Height Packing for Layered Manufacturing using an Extended Pattern Search Algorithm," ASME Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Chicago, Illinois, September 2-6, 2003.

Howie Choset, Kevin M. Lynch, Seth Hutchinson, George Kantor, Wolfram Burgard, Lydia E. Kavraki, and Sebastian Thrun, Principles of Robot Motion: Theory, Algorithms, and Implementations. 2005.

Coin3D. 2019. https://github.com/coin3d

Detroit DD13 Engine, https://demanddetroit.com/engines/dd13

Quan Ding and Jonathan Cagan, "Automated Trunk Packing with Extended Pattern Search," SAE Technical Paper Series, 2003-01-0671.

Dong, H., Guarneri, P., & Fadel, G. (2011). Bi-level approach to vehicle component layout with shape morphing. Journal of Mechanical Design, 133(4).

Friedrich Eisenbrand, Stefan Funke, Andreas Karrenbauer, Joachim Reichel, and Elmar Schomer, "Packing a Trunk - Now with a Twist!," SPM 2005 – ACM Symposium on Solid and Physical Modeling, Cambridge, Massachusetts, June 13-15, 2005.

Gottschalk, S., Lin, M., Manocha, D., Larsen, E.: Pqp–the proximity query package. http://gamma.cs.unc.edu/SSV/ (1999)

Pierre M. Grignon, John R. Wodziak, and Georges M. Fadel, "Bi-Objective Optimization of Components Packing using a Genetic Algorithm," AIAA, NASA, and ISSMO, Symposium on Multidisciplinary Analysis and Optimization, Bellevue, Washington, September 4-6, 1996.

Reginaldo Guirardello and Ross E. Swaney, "Optimization of Process Plant Layout with Pipe Routing," Computers and Chemical Engineering, vol. 30, no. 1, pp. 99-114, 2005.

W. Hills and N. Smith, "A New Approach to Spatial Layout Design in Complex Engineered Products," International Conference on Engineering Design, Tampere, Finland, August 19-21, 1997.

J. Hur, K. Lee, J. Ahn, and H. C. Lee, "A Three-Dimensional Algorithm using Two-Dimensional Slice Data for Building Multiple Parts in Layered Manufacturing," Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture, vol. 214, no. 5, pp. 365-378, 2000.

Ilkka Ikonen, William E. Biles, Anup Kumar, and Rammohan K. Ragade, "Concept for a Genetic Algorithm for Packing Three Dimensional Objects of Complex Shape," Proceedings of the First Online Workshop on Soft Computing, Nagoya University, August 19-30, 1996.

Ilkka Ikonen, William E. Biles, Anup Kumar, Rammohan K. Ragade, and John C. Wissel, "A Genetic Algorithm for Packing Three-Dimensional Non-Convex Objects having Cavities and Holes," Proceedings of the 7th International Conference on Genetic Algorithms, East Lansing, Michigan, July 19-23, 1997.

Ilkka Ikonen, William E. Biles, James E. Lewis, Anup Kumar, and Rammohan K. Ragade, "GARP: Genetic Algorithm for Part Packing in a Rapid Prototyping Machine," Intelligent Systems in Design and Manufacturing, Boston, Massachusetts, November 2-4, 1998.

Isight. 2019. https://www.3ds.com/products-services/simulia/products/isight-simulia-execution-engine/

Dohoy Jung and Dennis N. Assanis, "Numerical Modeling of Cross Flow Compact Heat Exchanger with Louvered Fins using Thermal Resistance Concept," SAE Technical Paper Series, 2006-01-0726.

KanGAL. 2019. https://www.iitk.ac.in/kangal/codes.shtml

Kang, N., Kokkolaras, M., Papalambros, P. Y., Yoo, S., Na, W., Park, J., & Featherman, D. (2014). Optimal design of commercial vehicle systems using analytical target cascading. Structural and Multidisciplinary Optimization, 50(6), 1103-1114.

Kim, H. M., "Target cascading in optimal system design," Ph.D. Dissertation, Mechanical Engineering Dept., University of Michigan, Ann Arbor, MI, 2001.

Kongsberg Oil & Gas Technologies. 2010. Coin3D - 3D Graphics Development Tools. https://github.com/coin3d/coin

Craig Lawrence, Jian L. Zhou, and Andre L. Tits, User's Guide for CFSQP Version 2.5. 1997.

Lee, K. J., 2014. Optimal System Design with Geometric Considerations. Ph.D. Dissertation, Mechanical Engineering, University of Michigan, Ann Arbor, Michigan, USA.

Lewis, K., Chen, W., & Schmidt, L. (2006). Decision making in engineering design. ASME press.

Mastinu, G. R. M., & Gobbi, M. (2001). On the optimal design of railway passenger vehicles. Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit, 215(2), 111-124.

Mathworks, 2019. https://www.mathworks.com/products/simulink.html

Michalek, J. J., Choudhary, R., and Papalambros, P. Y., "Architectural Layout Design Optimization". Engineering Optimization, Vol. 34, No. 5, 2002, pp. 461-484.

Michalek, J. J. and Papalambros, P. Y., "Interactive Design Optimization of Architectural Layouts". Engineering Optimization, Vol. 34, No. 5, 2002, pp. 485-501.

MPK, 2019. http://ai.stanford.edu/~mitul/mpk/

NX, 2019. https://www.plm.automation.siemens.com/global/en/products/nx/

Optimus, 2019. https://www.noesissolutions.com/our-products/optimus

Sungjin Park, Andreas Malikopoulos, Dohoy Jung, and Michael Kokkolaras, "Modeling and Optimization of Thermal Systems," Annual Report, University of Michigan, Ann Arbor, Michigan, 2007.

Sungjin Park and Dohoy Jung, "Numerical Modeling and Simulation of the Vehicle Cooling System for a Heavy Duty Series Hybrid Electric Vehicle," SAE Technical Paper Series, 2008-01-2421.

Sungjin Park and Dohoy Jung, "Design of Vehicle Cooling System Architecture for a Heavy Duty Series-Hybrid Electric Vehicle using Numerical System Simulations," Journal of Engineering for Gas Turbines and Power, Transactions of the ASME, vol. 132, no. 9, p. 092802 (11 pp.), 2010.

Papalambros, P. Y., & Wilde, D. J. (2017). Principles of optimal design: modeling and computation. 3rd Edition, Cambridge university press.

Peters, D. L., 2010. Coupling and Controllability in Optimal Design and Control. Ph.D. Dissertation, Department of Mechanical Engineering, University of Michigan, Ann Arbor, Michigan, USA.

POP Collision Detect. 2019. https://www.sidefx.com/docs/houdini/nodes/dop/popcollisiondetect.html

Sunand Sandurkar, Wei Chen, and Georges M. Fadel, "GAPRUS: Three-Dimensional Pipe Routing using Genetic Algorithms and Tessellated Objects," ASME Design Engineering Technical Conferences, Sacramento, California, September 14-17, 1997.

Segeborn, J., Carlson, J. S., Bohlin, R., & Söderberg, R. (2007, July). Geometry Design Supported by Minimizing and Visualizing Collision in Dynamic Packing. In Proceedings of world academy of science, engineering and technology, Vol. 25, pp. 164-171.

S. Szykman and J. Cagan, "Synthesis of Optimal Nonorthogonal Routes," Journal of Mechanical Design, Transactions of the ASME, vol. 118, no. 3, pp. 419-424, 1996.

S. Szykman, Jonathan Cagan, and P. Weisser, "An Integrated Approach to Optimal Three Dimensional Layout and Routing," Journal of Mechanical Design, Transactions of the ASME, vol. 120, no. 3, pp. 510-512, 1998.

Santosh Tiwari, Georges Fadel, and Peter Fenyes, "A Fast and Efficient Compact Packing Algorithm for Free-Form Objects," ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Brooklyn, New Work, August 3-6, 2008.

von Wysocki, T., Chahkar, J., & Gauterin, F. (2020). Small Changes in Vehicle Suspension Layouts Could Reduce Interior Road Noise. Vehicles, 2(1), 18-34.

Su Yin and Jonathan Cagan, "Extended Pattern Search Algorithm for Three-Dimensional Component Layout," Journal of Mechanical Design, Transactions of the ASME, vol. 122, no. 1, pp. 102-108, 2000.

Su Yin, Jonathan Cagan, and Peter Hodges, "Layout Optimization of Shapeable Components with Extended Pattern Search Applied to Transmission Design," Journal of Mechanical Design, Transactions of the ASME, vol. 126, no. 1, pp. 188-191, 2004.

Thomke, S., & Fujimoto, T. (2000). The effect of "front-loading" problem-solving on product development performance. Journal of Product Innovation Management, 17(2), 128-142.

Y. H. Yin, C. Zhou, and J. Y. Zhu, "A Pipe Route Design Methodology by Imitating Human Imaginal Thinking," CIRP Annals - Manufacturing Technology, vol. 59, no. 1, pp. 167-170, 2010.

Chen Zhou and Yuehong Yin, "Pipe Assembly Planning Algorithm by Imitating Human Imaginal Thinking," Assembly Automation, vol. 30, no. 1, pp. 66-74, 2010.

David Zhu and Jean-Claude Latombe, "Pipe Routing = Path Planning (with Many Constraints)," Proceedings of the 1991 IEEE International Conference on Robotics and Automation, Sacramento, California, April 9-11, 1991.