

A Knowledge-based Intelligent Diagnose Method for Pipe-routing

Jing Gong, Jianhua Liu, Boxuan Zhao

Key Laboratory of Fundamental Science for Advanced Machining Beijing Institute of Technology, Beijing 100081.China
gongjingbit@126.com, feng8181@bit.edu.cn

Abstract - Pipe-routing is a complex work especial in the fields of aerospace and shipbuilding because there are many engineering rules needed to be taken into consideration. This study presents an evaluating method based on engineering knowledges which is differ from traditional KBE, it makes the work more efficient and improve the reliability of pipe system as well.

Index Terms - Pipe-routing, Knowledge base, Answer Set Programming, Intelligent diagnose

I. Introduction

Pipeline system is widely used in complex products, which has an important influence to products' performances and reliabilities. Pipes are mainly used as the transmission carrier of the fuels, lubricants and coolant liquids in aerospace, automobile, petroleum and chemical industry. Its importance is equivalent as the blood vessels to humans. General Electric Corporation found that nearly 50% of the in-flight shutdown accidents were caused by the failures of the aircrafts' external pipelines and sensors [1]. Therefore, pipe-routing is extremely important in product's design process.

Pipe-routing is not only restricted by the narrow spaces of engine cases, but also restricted by the following rules [2]:

- 1) Avoidances of obstacles and engine accessories.
- 2) Minimum equipment clearance requirements.
- 3) Accessibility of valves by hand or by reach-rods.
- 4) Maximization of support sharing with other pipelines.
- 5) Minimization of pipeline length and number of bends.

In general, the difficulties of pipe-routing lie in three aspects: Firstly, pipe-routing is mainly completed in three-dimensional rotary space, and the product's integration is very high, which leads to narrow feasible space, besides, there are many accessories, which differ from each other in dimensions and shapes. Secondly, pipe-routing is restricted by many rules and all of them must be satisfied. Thirdly, the number of pipes in one engine is remarkable large (about 200-250 pipes in one typical engine) and usually they are also differ from each other [3].

Pipe-routing was traditionally done by eye, so it has badly delayed the production cycle [2]. In order to cut down costs and liberate human workers from such dull and irksome work, some CAD computer software emerged having the capability to do the works for humans automatically. But these software still have two fatal defects:

- 1) They merely concern about pipe's direction and pose, failing to connect engineering constraints and design principles with its geometric design.
- 2) The lack of diagnosis of designed pipelines results in low

reliability.

- 3) The work of design and check are carried out in different software involving the format conversion of the files which wastes too many time.

In order to make up CAD software's defects and liberate pipeline designers from their backbreaking labors so as to pay more attention to products' innovation design, we suggest applying the knowledge base technique of artificial intelligence into pipe-routing work.

II. Traditional KBE in product design

A. Definition

KBE means knowledge based engineering, which can solve engineering problems using artificial intelligence methodology and computer tools [4], and it combines product's engineering knowledge with their design and manufacturing process, with the ability to capture and systematically reuse product and process engineering knowledge, and the final goal is to reduce products' development costs by means of giving advises to designers who are short at design experiences supporting multidisciplinary design optimization in all the phases of product design process.

B. Key techniques

There are three technique problems when we are molding knowledge-based systems; they are knowledge acquisition, knowledge representation and knowledge reasoning [5]. The last two have become the bottle-neck in the process of develop knowledge-based systems.

Knowledge acquisition: Knowledge acquisition usually means knowledge engineers collaborate with domain experts to extract and summary their experiences and expertise.

Knowledge representation: Knowledge representation means the formalization of expertise, which select a kind of data structure that computer can understand to express expert knowledge. With the development of AI, there have been several kinds of methodologies such as: the first-order predicate logic; production; frame and semantic network. In this paper we select the first-order predicate logic to express engineering knowledge.

Knowledge reasoning: Knowledge reasoning is the process of problem solving and machine thinking using formalized knowledge depended on the method of knowledge representation. There are forward reasoning, backward reasoning and bidirectional reasoning from the point of

reasoning direction. Besides, there are monotonic reasoning and non-monotonic reasoning distinguished by the relationship between the account of information and time. And in this paper we adopt the non-monotonic reasoning.

C. The reasons we abandon traditional KBE

Although the use of KBE in CAD has raised the productivity and liberated workers from heavy and irksome drawing works, but it has little practical value in pipe-routing because the KBE system can only be used in the modelling process of relatively simple and independent parts. But the environment of pipe-routing is always complex, pipes designed latter are obviously influenced by the former ones. Besides, most of pipe-routing knowledge is ambiguous, and they can't be understood well by KBE systems. Therefore, we developed a new tool called "intelligent pipe-routing diagnose system" to solve such problems. But at first, let's see the procedure and constraints.

III. Procedure and constraints of pipe-routing

A. Procedure

The procedure of pipe-routing is shown in Fig.1, as we can see, the work is very complicate because it must deal with multidisciplinary design optimization involving structure, dynamics and statics. Besides the check of pipes' parameters is carried out in different software, so it usually results in severe manufacturing and assembly problems and no wonder the ratios of rework is always high.

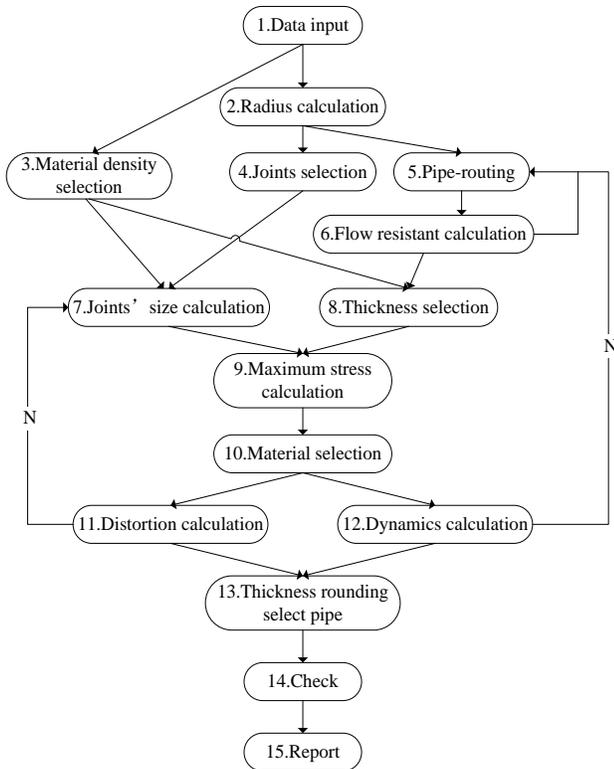


Fig.1 The procedure of pipe-routing

B. Understanding constraints

Pipe-routing must comply with the following engineering rules:

Manufacturability: the designed pipes cannot exceed the manufacturing ability of CNC pipe-bending machine.

Simple and Compact structure: material and installation costs are mainly tied to the length and the number of bends of pipelines in spite of components and accessories, so they are needed as little as possible in condition of satisfying pipe system's function.

Auxiliary supporting: there must need some auxiliary supporting except the welding in the joints considering the pipe's weight and vibration of the engine.

Maintainability: a designed pipeline should be easily installed and disassembled within worker's arm's reach.

Dynamics rules: pipe designers must concern about sympathetic vibration factor. Pipe's natural vibration frequency needs to be adjusted by $f_n \geq 1.25f_{max}$

or $f_n \leq 0.8f_{idl}$. Among of it f_n is pipe's natural vibration frequency, f_{max} is engine's max speed and f_{idl} is engine's idle speed. Besides the fatigue strength coefficient of the welding point must satisfy the formula: $k_v = \sigma_a / \sigma_v \geq 3$.

σ_a is the pipe's fatigue strength and σ_v is the max alternating stress of the engine's joints and supports in working state. Otherwise, the total resistance loss is the value of on-way resistance added with local resistance which is calculated as $c \cdot f_i$ —friction factor determined by experiment, ξ —local resistance factor, l_i —the length of pipe, d —the diameter, ρ —density of medium, v_i —velocity of the flow.

Statics rules: the pipe's bursting pressure is calculated as $p_b = \sigma_b [(\frac{d}{\delta} + 1) / (\frac{1}{2}(\frac{d}{\delta})^2 + \frac{d}{\delta} + 1)] \cdot \sigma_b$ —the strength limit of the material, d —pipe's external diameter, δ —the thickness. And the permitted maximum pressure is calculated as: $p_{gm} = p_b / k_b$. p_b —bursting pressure calculated as above, k_b —the strength reserve coefficient of specific material.

Engineering aesthetics rules: the appearance of the pipe system should be artistic in the point of industrial design.

IV. Implementation of the intelligent evaluating system

A. Modeling of knowledge base

As I had mentioned above, knowledge representation and reasoning are the key techniques of AI, and in order to represent engineering knowledge accurately, we select a new methodology called "Answer Set Programming (ASP)" which is the cross merging of logic program and non-monotonic reasoning [7]. Logic program is constructed with facts and rules. In this paper, facts refer to the information of designed pipes which is shown in Fig.2, and rules mean the engineering

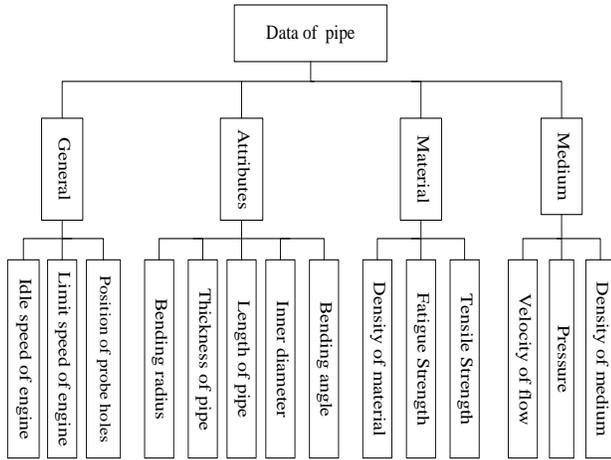


Fig.2 Data of the designed pipes

knowledge in pipe-routing. Thus, logic program can be expressed as:

$$\text{Logic program} = \text{Facts} + \text{Rules}$$

The minimum unit of the ASP program's sentence is an item which can be a constant, a variable or a functional word. And a sentence is constructed with items which are connected by the following symbols as Table I shown, and their truth-table is shown in Table II.

TABLE I Connecting Symbols for Logic Items

| | |
|---|-------------------|
| → | Logic implication |
| ∧ | Logic conjunction |
| ∨ | Logic disjunction |
| ¬ | Logic negation |

TABLE II The truth-table of connecting symbols

| P | Q | P∨Q | P∧Q | ¬P | P→Q |
|---|---|-----|-----|----|-----|
| T | T | T | T | F | T |
| F | T | T | F | T | T |
| T | F | T | F | F | F |
| F | F | F | F | T | T |

A normal logic program is the finite set of normal rules, and a normal rule has the following form:

$$H \leftarrow a_1, \dots, a_m, \text{not } a_{m+1}, \dots, \text{not } a_n$$

$n \leq m \leq 0$, $a_1 \dots a_n$ are atoms, and the word "not" means negation as failure. H is either an empty set or an atom, if H is an empty set, then this rule is called constraint, else it's called proper rule. And if $n = m = 0$, then it becomes a fact. If all of the rules don't contain the word "not", then we call this logic program as positive program [7].

Then I will show you how the engineering knowledges are expressed in ASP specification; take the next knowledge as an example:

A designed pipe whose id number is 8024 and its external diameter is 50mm, the bending radius is 120mm, and there is another designed pipe whose id number is 8459, and its external diameter is 60mm, the bending radius is 135mm, the clearance is 4mm and the allowed minimum clearance between the two pipes is 6mm.

We need three steps to express this knowledge in ASP specification. First, we need to define the predicates, for example we define predicates "pipe(X)" means X is a pipe, "diameter(X, D)" means the diameter of X is D, "radius(X, R)" means the bending radius of X is R, "gap(X, Y, C)" means the clearance between X and Y is C, "mingap(G)" means the allowed minimum clearance. Second, we need to replace the variables with values like: pipe (8024), diameter (8024, 50). Third we need to connect these predicates with the connecting symbols as Table I shown. Then we get the logic program:

- 1) $\text{pipe}(8024) \wedge \text{diameter}(8024, 50) \wedge \text{radius}(8024, 120)$.
- 2) $\text{pipe}(8459) \wedge \text{diameter}(8459, 60) \wedge \text{radius}(8459, 135)$.
- 3) $\text{gap}(8024, 8459, 4) \wedge \text{mingap}(6)$.
- 4) $:- \text{mingap}(C) \wedge \text{gap}(ID1, ID2, \text{value}) \wedge \text{value} < C$.

After completed the logic program, then we just give it to the "ASP" solver. And the solving process can be divided into two steps, the first step is "grounding" which replaces all of the variables in logic program with specific values, and the other is "model search" which calculates the "answer set" of the logic program grounded in the first step. And if we have expressed all the data of designed pipes as Fig.2 shown into logic program's format, then the modeling of the knowledge base is completed.

B. Example verification

Now I will introduce you how the evaluating system works, and the procedure is shown in Fig.3. Firstly, we do the pipe-routing work in CAD software like UG and PRO/E. Secondly; we export the data of designed pipes in XML format. Thirdly, we add engineering knowledges into the knowledge base and use them to evaluate the routed pipelines to see whether they are acceptable or not. Fourthly, we import the XML exported from the evaluating system into CAD software and highlight the defective pipelines and redesign them.

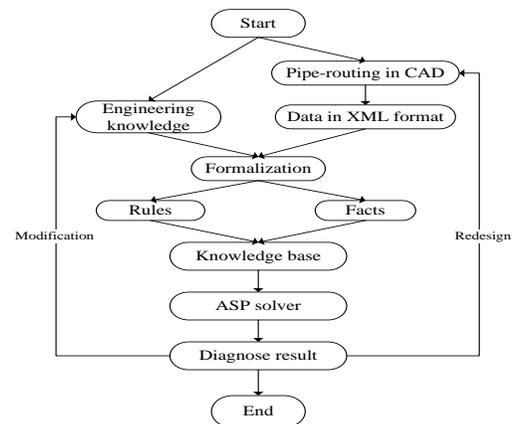


Fig.3 Procedure of the diagnose system

In this paper, we do the pipe-routing work in VAPP (a virtual assembly platform which can be used for pipe-routing) at first as Fig.4 shown. And then we call the diagnose system using the XML file exported from VAPP as the input. In this step, we can see which engineering rule is not satisfied and where the defect happens directly from the diagnosis results as Fig.5 shown. Finally, VAPP can highlight the unacceptable pipelines, and it's very convenient for pipe-routing designers to amend them as Fig.6 shown.

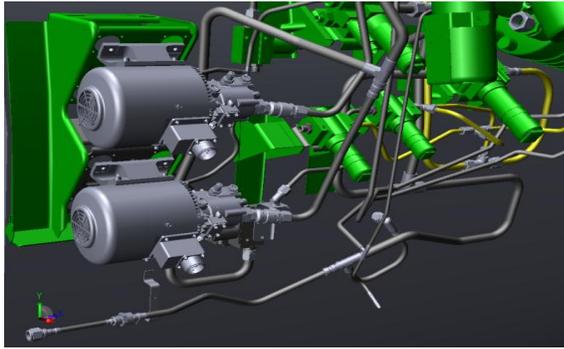


Fig.4 Pipe-routing in VAPP

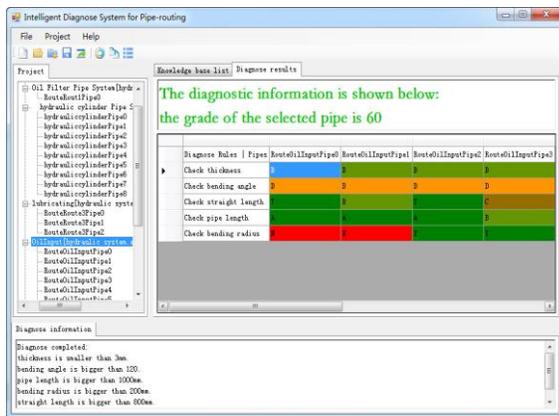


Fig.5 Diagnose results

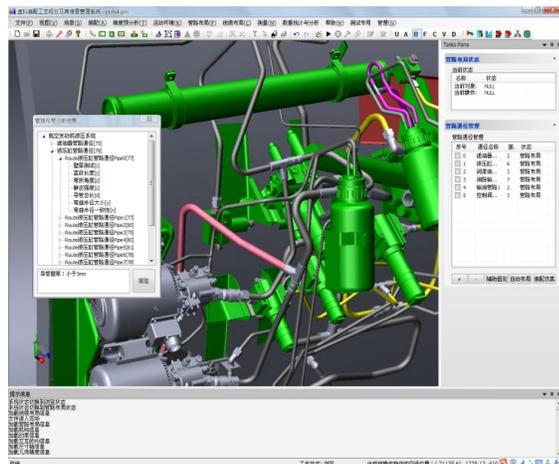


Fig.6 Highlight the defective pipe in VAPP

V. Conclusion

In this paper, the authors analyzed the problems lie in traditional pipe-routing first, and then a diagnose system for pipe-routing was developed. They use a new knowledge representation and reasoning method which combines engineering rules with pipe's geometric design successfully. And the knowledge base of this diagnose system is flexible that makes engineers can add and edit the rules easily. With its use in pipe-routing, the production cycle will be shortened largely, and more important, pipes' reliability will be improved. So we can imagine that it will have a good application prospect in the future.

Acknowledgment

The authors would like to acknowledge the Digital Design and Manufacturing Laboratory of Beijing Institute of Technology for providing pipe-routing software (VAPP), and they also would like to thank Hai Wan, Junqiang Wen, Zhanhao Xiao, Jinhui Zhang and Wei Huang in Sun Yat-Sen University for providing ASP solver and valuable technical advice.

References

- [1] Q. Liu, C. Wang, and X. Bai, "Engineering Rules-based Pipe Routing Algorithm for Aero-engines", JOURNAL OF MECHANICAL ENGINEERING, vol. 47, no. 5, pp. 163-168, March 2011.
- [2] J. Park, R. Storch, "Pipe-routing algorithm development: case study of a ship engine room design", Expert Systems with Applications, no. 23, pp. 299-309, 2002.
- [3] Z. Lu, R. Ning, J. Liu, and B. Wan, "Pipe layout technology in virtual environment", Computer Integrated Manufacturing Systems, vol. 14, no. 8, pp. 1484-1488, August 2008.
- [4] B. Zhao, R. Ning, J. Liu, "Pipe Layout and Assembly Technology in Virtual Environment", Transactions of Beijing Institute of Technology, vol. 30, no. 8, pp. 895-900, August 2010.
- [5] J. Chen, H. Yang, R. Jiang, and D. Wang, "Application of knowledge based engineering methods for hull structural member design", SHIP SCIENCE AND TECHNOLOGY, vol. 32, no. 10, pp. 16-20, October 2012.
- [6] C. Wang, Q. Liu, X. Bai, and J. Wang, "Pipe routing for aero-engines in complex constraint space", Computer Integrated Manufacturing Systems, vol. 16, no. 11, pp. 2327-2332, November 2010.
- [7] J. Ji, "Several Ways to Improve Efficiency of Answer Set Programming and Its Application in Service Robots", University of Science and Technology of China, March 2010.
- [8] G. Rocca, "Knowledge based engineering: Between AI and CAD. Review of a language based technology to support engineering design", Advanced Engineering Informatics, no. 26, pp. 159-179, 2012.
- [9] X.F. Zha, H.J. Du, "Knowledge-based approach and system for assembly oriented design", vol. 12 no. 14, pp. 61-75, 2001.
- [10] P. Cunningham, A. Bonzano, "Knowledge engineering issues in developing a case-based reasoning application", Knowledge-Based Systems, vol. 12, no. 7, pp. 371-379, November 1999.