Western Multiconference (WMC 98) featuring
Int. Conf. on Simulation and Multimedia in Engineering Education (ICSEE 98).
San Diego, California, 92177 U.S.A., 1998. Eds.: D. Davani and D. Elizandra.
pp. 274-279, ISBN 1-56555-142-7.

Enriching Engineering Education in Fluidics

Benno Stein   Daniel Curatolo   Marcus Hoffmann

Department of Mathematics and Computer Science—Knowledge-based Systems
University of Paderborn, D–33100 Paderborn, Germany
email: stein@uni-paderborn.de

Keywords: computer-aided design, computer-aided analysis, education, interactive simulation

Abstract

Engineering education in fluidics means to convey the physical principles behind pneumatic or hydraulic building blocks such as valves or cylinders. Up to now fluidic hardware plays an important role in this connection: Education laboratories with semi-professional components are used to build up small circuits and to perform experiments respecting hydraulic and pneumatic laws of nature.

Based on research and experiences in the fluidic engineering domain we have realized tools for drawing, simulating, and behavior visualization of electro-fluidic circuits. These tools aim at a core objective within the fluidic education: gathering experience by (riskless) experimenting with fluidic and electric building blocks, thus providing an immediate feedback when trying to answer “what-if”-questions.

This paper outlines selected concepts of our tools as well as of our current research in this field.

1 Introduction

In first place engineering education in fluidics means to convey the physical principles behind basic hydraulic and pneumatic building blocks, such as valves, cylinders, or supply elements. As well as that students in this field must acquire synthesis know-how to become able to design circuits on their own; i.e., they have to learn in which way fluidic building blocks are combined and connected to set up a hydraulic or pneumatic system that fulfills given requirements.

Till now fluidic hardware plays an important role in this connection: Education laboratories with semi-professional components are used to build up small circuits and to perform experiments respecting hydraulic and pneumatic laws of nature.

Fluidic education can be simplified by means of software, it can be made more cost-effective, and—not least, there is the chance to teach additional and more complex aspects from this field.

In the last couple of years and throughout different projects we have been developing concepts to support the analysis and the design of fluidic systems. Among others these concepts include standard numerical methods and algorithms from graph theory, which have been adapted and improved with knowledge-based concepts or domain knowledge.

Based on our experience we have realized the electro-hydraulic and electro-pneumatic education tools Fluid*SIM*-H and Fluid*SIM*-P respectively. These tools aim at a core objective within the fluidic education: gathering experience by riskless experimenting with fluidic building blocks, thus providing an immediate feedback, e. g. when trying to answer “what-if”-questions.

Our Fluid*SIM* tools are not intended to replace the human instructor—but to complement and to enrich education. They establish integrated environments for CAD-like drawing, modeling, simulation, and behavior visualization of circuits along with a close integration of didactics material such as exercises, animated illustrations, photos, and films[1]. Due to both their appealing handling on the one hand as well as their powerful simulation capabilities on the other, these tools found a broad acceptance in fluidic engineering education.[2]

---

[1] The concepts and the integration of these didactics materials is not a matter of the paper in hand.

[2] At present Fluid*SIM* (hydraulics and pneumatics) has been translated into English, German, and Spanish; it runs under Windows (3.1, 95, NT) and is distributed by the Festo Didactic GmbH & Co., Esslingen, Germany.

The next section gives an idea of how to work with Fluid*SIM* using it as a fluidic building kit, e. g. when designing circuits. The sections 3 and 4 outline underlying concepts and current research respecting different fluidic analysis tasks.

2 Learning Circuit Design with Fluid*SIM*

In reality, circuit design as carried out by a designer happens within the following steps:

1. Having interpreted the given demands, a first solution is sketched out by drawing a simplified circuit.
2. The draft circuit is analyzed by checking syntactical, geometrical, logical, and dimensional constraints. Typical examples are open pipes, wrong connections, the switching logic, orders of magnitude, or parameter ranges.
3. According to the analysis result the current design is modified and refined.

Step 2 and 3 are repeated until the circuit fulfills all demands.

This procedure can be experienced and varied easily using Fluid*SIM*. I. e., aside from studying physical dependencies, a user can also learn the projecting procedure in fluidic engineering.

In Fluid*SIM*, components are selected, arranged, connected, dimensioned, and simulated while the model formulation process is made transparent: The information that is necessary for the checking and simulation process is derived from the drawing. E. g. while drawing a line between two components' gates the appropriate pipes are instantiated; during simulation Fluid*SIM* detects, schedules, and processes events caused by discontinuous component state changes such as from relief valves that may open or shut. Figure 1 and 2 depict snapshots when working in Edit and Simulation mode respectively.

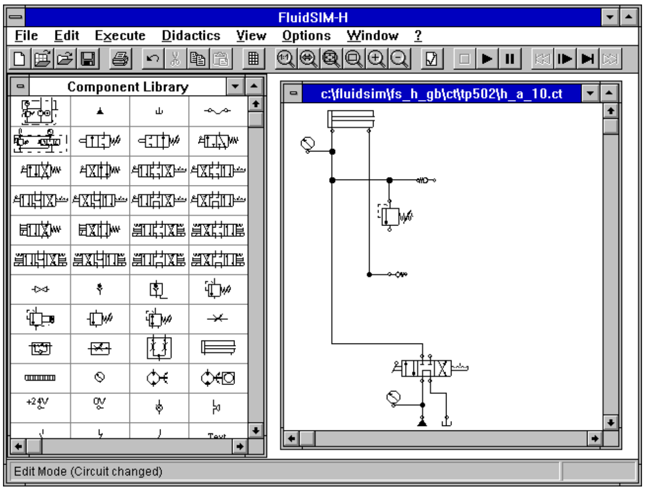

Figure 1: The snapshot shows a part of the component library and a circuit currently edited.

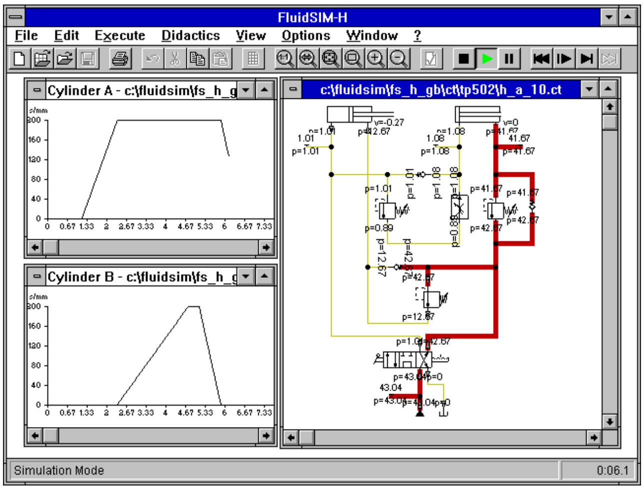

Figure 2: This snapshot shows a simulation run and the cylinders' related distance/time diagrams.

Fluid*SIM* provides for a small electric component library to build control setups. So, given a circuit containing electro-fluidic components and sensors, electric control setups can be used to control the fluidic circuit by processing and generating events. Figure 3 shows an example.

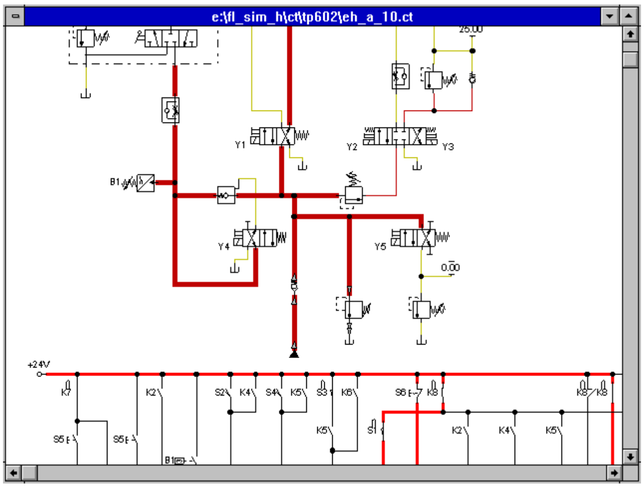

Figure 3: Fluidic circuit coupled with an electric control circuit.

During simulation also the user is allowed to trigger events by operating components like switches or valves. The related models are updated immediately in the background, thus providing the feeling of interacting with a running system.

3 Circuit Analysis

From a physical standpoint, Fluid*SIM* realizes a stationary behavior simulation, which is based on nonlinear component models. The term "stationary" relates to the pressure, p, and, equivalently, to the flow, Q, and the first derivative of the piston position, $\dot{x}$. An implication of this simulation approach is that two classes of events are to be distinguished, namely events that must be skipped and events that are to be obeyed.

Events of the first class are triggered either by physical thresholds or by *reactive* state changes of particular components such as relief valves. These events can occur only if $\dot{p} \neq 0$ or if $\dot{Q} \neq 0$. Events of the second class are triggered by *active* state changes of components like cylinders, directional valves, and switches, or by the user himself.

Hence, to directly compute the desired stationary behavior of a fluidic circuit, assumptions that reason about component states must be made. The next subsection presents an illustrating example.

3.1 Example

Given is a circuit as drawn in Figure 4, consisting of three cylinders with different loads, three pressure relief valves, `rv_1`, `rv_2`, and `rv_3`, and the necessary supply elements. The task is to determine (*i*) which of the cylinders will extend if the pump is switched on, and (*ii*) the pressure p_x, at the bottom of the rightmost cylinder. The additional values, such as hydraulic resistances, pressure thresholds, geometrical values, etc. are given.

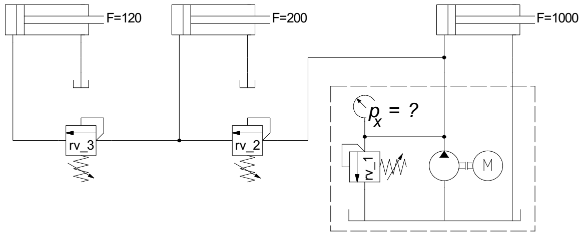

Figure 4: Circuit with three coupled cylinders.

Figure 5 shows the first stationary state of the circuit after the pump has been switched on: Among others, the rightmost cylinder extends, and all relief valves are closed. Exactly this situation is directly envisioned by Fluid*SIM*. By the way, the second stationary state is reached when the piston of the right cylinder hits the stop and, as a consequence, triggers an event.

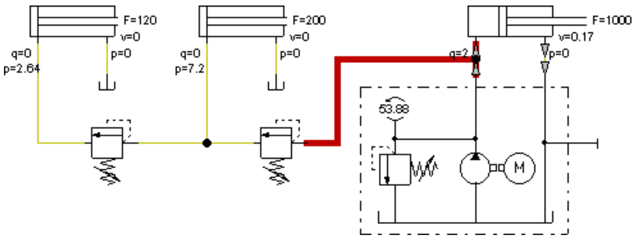

Figure 5: First stationary state after pump is switched on.

Figure 6 takes a closer look on what has happened until the first stationary state is reached at the time t_6. In fact, while the relief valve `rv_1` remains closed when the pump is switched on, both relief valves `rv_2` and `rv_3` open and close, resulting in pressure drops as well as stalled pressures between the middle cylinder and `rv_2` and the left cylinder and `rv_3` respectively. Altogether five reactive state changes may occur til the first stationary time interval is entered, presumed an average detailed dynamic simulation.

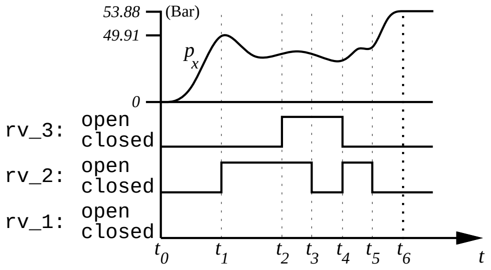

Figure 6: A look at the dynamical development of p_x.

Of course all stationary states can be found by tracing the derivatives of the state variables, but there are crucial points bound up with the necessary dynamic simulation: efficiency requests, trade-offs between correct behavior and model precision, issues of transparency and expressiveness relating the analysis, and, not least, the complexity regarding educational objectives. In this place we will not pick up a discussion on this thread but concentrate on the realization of a stationary state analysis.

Figure 7 shows the state space of the three relief valves from our example. Here, m_{rv_a} and m_{rv_b} stand for the behavior models of a closed and open relief valve respectively; e. g. the set $\{m_{rv1_a}, m_{rv2_a}, m_{rv3_a}\}$ states that all valves are closed, which corresponds to a correct state assignment for the first stationary state at t_6 in the example. There exist 8 state combinations relating the relief valves.

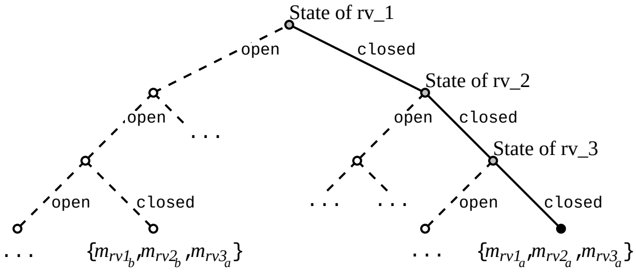

Figure 7: State space of the three relief valves.

Also the cylinders provide for two distinct states—(*i*) an equilibrium state, m_{cyl_a}, where the balance of forces holds between the pressure, $\Delta p \cdot A$, the load, F, and the piston velocity, $k \cdot \dot{x}$, and (*ii*) a stop state, m_{cyl_b}, where the piston touches one side or other of the cylinder.

At time t_6 in the example, the extending cylinder is in state m_{cyl_b} while the others are in state m_{cyl_a}. Hence, the tuple $\{m_{rv1_a}, m_{rv2_a}, m_{rv3_a}, m_{cyl1_b}, m_{cyl2_a}, m_{cyl3_a})$ completely defines the state space in the first stationary state. The total state space of the circuit contains $2^6 = 64$ elements.

We call the construction of the correct global state vector *model synthesis*: Local component models are synthesized towards a global behavior model.

To solve the model synthesis problem, that is to say, to find a correct state vector, it is usually not necessary to evaluate all possible state combination. A large part of the state space can be cut, if dependencies between particular states are exploited, or if domain knowledge or heuristics guide the search.

E. g., in the circuit of Figure 4 all state vectors containing tuples of the form $\{m_{rv\,n_a}, m_{cyl\,n_b}\}$ or $\{m_{rv\,n_b}, m_{cyl\,n_a}\}$, $n \in \{2, 3\}$, can be discarded at the outset: Cylinder 2 (3) can only extend if `rv_2` (`rv_3`) is open. Tuples that define *physically* contradictory state combinations are called *nogoods*.[3]

The next subsection further elaborates on model synthesis.

3.2 Model Synthesis

Even though a circuit diagram C has a useful and definite physical interpretation, its mathematical description cannot be derived in an ad-hoc manner: Most components of C are defined by a *set* of behavior constraints from which the relevant ones must be selected. This problem, called model synthesis[4] here, consists of all steps that are necessary to set up a global behavior model which is both correct in a physical sense and *locally unique* (cf. [Stein, 1998]).

The indeterminacy of local behavior descriptions originates from the following causes:

- *Component States.* Most components have different physical states, each coupled with a particular behavior description. The actual validity of a state depends on the entire system and the actual input parameters. Example: A pressure relief valve may be in the state “opened” or “closed”.
- *Topology.* A hydraulic system’s topology can change with a component’s state. Example: Depending on its switching position a proportional valve connects different parts of a hydraulic network.
- *Physical Thresholds.* Even for a fixed state the direction or the absolute value of a physical quantity, which is a-priori unknown, may cause different behaviors of a component. Example: A turbulent flow is described by another pressure drop law than a laminar flow.

Precisely stated, for each component o in C let $M_o = \{m_{o_1}, m_{o_2}, \ldots, m_{o_k}\}$ be comprised of the k behavior alternatives of o. If a component o has a locally unique model, say, a pipe for instance, $|M_o| = 1$. Let $\mathcal{M}_C$ be the Cartesian product of the sets $M_o, o \in C$. $\mathcal{M}_C$ comprises the possible global models of the circuit C, and thus, $\mathcal{M}_C$ defines the total synthesis search space.

To reason about the behavior constraints m_{o_i} of a component o, some kind of meta constraints, the so-called model selection constraints, are needed. Example:

```
IF $x$ is of type relief_valve
  AND $x$ is in state open
  THEN $m_{rv_a} := \{Q_A = Q_B, \ldots\}$ is valid
```

The `IF`-clause constitutes a model selection constraint, m_{rv_a} is one of the local behavior models of the relief valve, and “$Q_A = Q_B$” is a part of the behavior constraint of m_{rv_a}. A model selection constraint is called active if its conditions are fulfilled.

Given the concept of model selection constraints, the search for a physically consistent model can be realized as a cycle containing the following steps:

1. *Component Selection.* Select a component that possesses several states, that is to say, behavior alternatives.
2. *State Selection.* Choose a state for this component.
3. *Model Synthesis.* Identify and evaluate active model selection constraints[5]. Synthesize the local behavior models to a global model M_C.
4. *Simulation.* Simulate the synthesized behavior model M_C by evaluating the behavior constraints.
5. *Modification.* In case of physical inconsistencies, trace back to a choice point, formulate additional synthesis restrictions, and set up a new M_C.

Figure 8 illustrates the search process graphically: The circuit C defines $\mathcal{M}_C$, from which a subset is selected ($= M_C$), simulated ($\Rightarrow B_C$, the behavior of C under M_C), and compared to $\mathcal{B}_{Hyd}$, which stands for the universal behavior laws of hydraulics.

The search comes to an end if either a consistent global behavior model is found or if no further choice point exists. Note that different components constrain the model synthesis process in a different manner. Hence, the order by which undetermined components are processed plays a crucial role.

A strength of Fluid*SIM* comes with its powerful concepts to efficiently solve the model synthesis problem. Aside from the deployment of domain knowledge, there are two mechanism to *automatically* create synthesis restrictions (nogoods) that cut down the search space: (*i*) a topological circuit analysis, (*ii*) a dependency recording based on state assumptions.

---

[3] There still exist further nogoods, such as $\{m_{rv\,n_b}, m_{rv3_a},\}$, $n \in \{2, 3\}$: `rv_2` or `rv_3` can be open only if `rv_1` is open as well.

[4] There is particular research in connection with model composition problems (cf. [Nayak, 1992; Falkenhainer and Forbus, 1991; Iwasaki and Levy, 1993]). Note that the mentioned as well as related work focuses on the construction or selection of adequate models with respect to different tasks (simulation, diagnosis) or different levels of granularity. This is not the case here: Although both the task and the level of granularity are given, there is a synthesis problem, which results from the indeterminacy of local behavior descriptions in the hydraulic domain.

[5] This type of inference is sometimes called “constraint inference”, as opposed to a “value inference” process that is performed during simulation, [Davis, 1987].

$$\begin{tikzcd}
C \arrow[d] & \\
\mathcal{M}_C = M_1 \times M_2 \times \ldots \times M_{|C|} \arrow[d, "{\text{Model selection, Model synthesis}}"] & \mathcal{B}_{Hyd} \arrow[l, "\text{Model modification}"'] \arrow[ddl, leftrightarrow, "\text{Physical evaluation}"'] \\
M_C \arrow[d, "\text{Simulation}"] & \\
B_C &
\end{tikzcd}$$

Figure 8: Exploring the synthesis search space $\mathcal{M}_C$.

Constraint processing in Fluid*SIM* encloses standard numerical procedures as well as inference methods for value propagation or rule processing. In this text will not engage into constraint processing details; additional information may found in [Curatolo, 1996; Stein, 1998], and [Stein, 1995].

4 Structure Envisioning

Taking the view of an experienced designer, a circuit is more than a collection of its components—it contains a functional structure. This functional structure is reflected by so-called hydraulic or pneumatic "axes" and the couplings between these axes. An axis realizes an independent subfunction of a fluidic system; e. g. within a hydraulic press all components that realize the ejector form together a single hydraulic axis. Figure 9 shows a few examples for hydraulic axes that have been cut free from a large circuit.

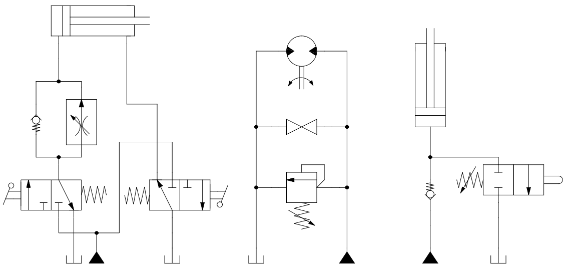

Figure 9: Examples for hydraulic axes.

The identification of the axes and couplings in complex circuits is rather difficult for novices, and consequently their visualization is of a great educational value. Together with domain experts we have developed algorithms that automatically envision the functional structure when a given a fluidic circuit[6].

The analysis procedure is comprised of the following steps:

1. *Graph-theoretical Formulation.* Starting point is an abstraction from a circuit C onto a simplified graph data structure $G_h(C)$. As subsection 4.1 shows, this data structure also forms the basis for a precise definition of the couplings between axes.
2. *Preprocessing.* To reduce G_h's complexity—but, in first place, to make axes identification possible at all, G_h is simplified by means of merging, deletion, and condensation rules.
3. *Axes Identification.* Identifying a hydraulic axis means to search for a set of nodes in the hydraulic graph whose counterpart in the circuit realizes a particular function. For the most part the identification of axes can be reduced onto the solution of shortest paths problems within in the preprocessed graph G_h.

The remainder of this section shows in which graph theory is employed to formulate fluidic connections. A detailed decription of the above procedure can be found in [Stein and Schulz, 1998].

4.1 Graph-theoretical Formulation of Fluidic Concepts

The topological analysis as pursued here is a matter of graph theory, and, in the following, we will fall back on some basic graph-theoretical terms such as *multigraph*, *path*, or *connected component*.[7]

A related *hydraulic graph* $G_h(C)$ of a circuit C is a multigraph $\langle V_C, E_C, g_C \rangle$ whose elements are defined as follows. (*i*) V_C is a set of points, and there is a bijective mapping from the set of non-pipe components in C onto V_C. (*ii*) E_C is a set of edges, and there is a bijective mapping from the set of pipe components in C onto E_C. (*iii*) $g : E_C \rightarrow 2^{V_C}$ is a function that maps an $e \in E_C$ onto $(v_i, v_j) \in 2^{V_C}$, if and only if there is a pipe between the components associated with v_i, v_j, and if e is associated with this pipe.[8]

Figure 10 depicts a circuit and its related hydraulic graph.

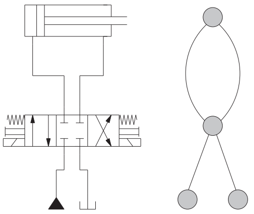

Figure 10: Circuit and its related hydraulic graph.

To accomplish complex manufacturing or manipulation tasks, several hydraulic axes are coupled and play together. Its in the nature of things that the level of such a coupling can vary, from rather loosely coupled axes to axes

---

[6] By now the topological analysis is not an integral part of Fluid*SIM*.

[7] We use these definitions in a standard way as specified in [Cormen *et al.*, 1990; Jungnickel, 1990], and [McHugh, 1990].

[8] We need multigraphs instead of graphs since components of a fluidic system may be connected in parallel.

that strongly depend on each other. Note that in order to determine those components of a circuit belonging to an axis A, all couplings between A and other axes must be identified as such.

In particular we distinguish between informational, parallel, series, and sequential couplings; graph theory provides a proper means to define these couplings—Example:

Given is a circuit C containing two sub-circuits A, B, which realize two different hydraulic axes. Let $G_h(A) := \langle V_A, E_A, g_A\rangle$ and $G_h(B) := \langle V_B, E_B, g_B\rangle$ denote the related hydraulic graphs of A and B respectively. Moreover, let $V'_A := V_A - (V_A \cap V_B)$, $V'_B := V_B - (V_A \cap V_B)$, and let $P_{w,s}$ be the set of all those paths from the working element w to a supply element s that use no edge associated with a control line. Then A and B are coupled in parallel if $\exists v_x \in V_X, X \in \{A, B\}$ such that the following conditions hold:

(*i*) v_x is associated with a control element.

(*ii*) $\forall p \in P_{w,s} \cap V'_x, p = (v_1, \ldots, v_n) : \exists i \in \{1, \ldots, k\}$ with $v_x = v_i$.

Figure 11 gives an example.

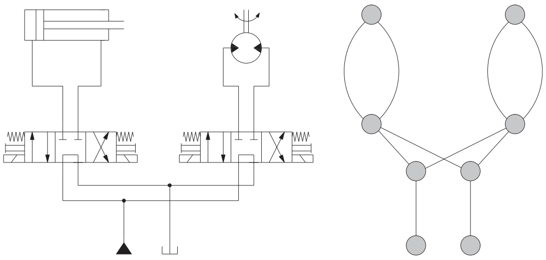

Figure 11: Circuit containing hydraulic axes coupled in parallel.

From an engineering point of view the former definition states that each of the axes A and B is controlled by its own control element. The other coupling types have been defined in a similar manner. Loosely speaking, fluidic axes are searched by applying these definitions to the graph G_h.

The algorithms for the topological analysis were evaluated with our circuit library, which contains more than 150 circuits at the moment. About 95% of the axes in these circuits can be identified correctly.

5 Conclusion and Future Work

The broad acceptance of our Fluid*SIM* tools has shown that they complement and enrich the education in fluidic engineering. Aside from the easy handling, one of the strong points of the tools are their simulation capabilities: Providing an immediate feedback while riskless experimenting with fluidic building blocks, Fluid*SIM* encourages students in posing “what-if”-questions.

To realize the Fluid*SIM* approach we have developed tailored concepts to support the analysis of fluidic systems. These concepts include standard numerical methods, algorithms from graph theory, and symbolic inference mechanisms, all of which have been adapted and improved with knowledge-based concepts or domain knowledge.

Our future work is twofold. Within educational respects we are working on the integration of new features such as structure envisioning or the coupling to external NC-hardware. Within professional respects, research concentrates on the operationalization of design knowledge within Fluid*SIM*.

References

[Cormen *et al.*, 1990] Thomas H. Cormen, Charles E. Leiserson, and Ronald L. Rivest. *Introduction to Algorithms*. The MIT Press, Cambridge. Massachusetts, 1990.

[Curatolo, 1996] Daniel Curatolo. *Wissensbasierte Methoden zur effizienten Simulation fluidtechnischer Systeme*. Dissertation, University of Paderborn, Department of Mathematics and Computer Science, 1996.

[Davis, 1987] Ernest Davis. Constraint Propagation with Interval Labels. *Artificial Intelligence*, 32:281–331, 1987.

[Falkenhainer and Forbus, 1991] Brain Falkenhainer and Kenneth D. Forbus. Compositional Modeling: Finding the Right Model for the Job. *Artificial Intelligence*, 51:95–143, 1991.

[Iwasaki and Levy, 1993] Yumi Iwasaki and Alon Y. Levy. Automated Model Selection for Simulation. Knowledge Systems Laboratory, Stanford University (QS93), 1993.

[Jungnickel, 1990] Dieter Jungnickel. *Graphen, Netzwerke und Algorithmen*. BI Wissenschaftsverlag, Wien, 1990.

[McHugh, 1990] J. McHugh. *Algorithmic Graph Theory*. Prentice Hall, 1990.

[Nayak, 1992] P. Pandurang Nayak. *Automated Model Selection*. Dissertation, Stanford University, 1992.

[Stein and Schulz, 1998] Benno Stein and André Schulz. Topological Analysis of Hydraulic Systems. Technical Report tr-ri-98-197, University of Paderborn, Department of Mathematics and Computer Science, July 1998.

[Stein, 1995] Benno Stein. *Functional Models in Configuration Systems*. Dissertation, University of Paderborn, Institute of Computer Science, 1995.

[Stein, 1998] Benno Stein. Supporting Hydraulic Circuit Design by Efficiently Solving the Model Synthesis Problem. *Proc. EIS 98, International ICSC Symposium on Engineering of Intelligent Systems, University of La Laguna, Tenerife, Spain*, February 1998.