



SIMULATION OF GRADIENT-BASED INDIVIDUAL DESIGN BEHAVIOUR IN DISTRIBUTED DEVELOPMENT PROCESSES

F. Wöhr¹, M. Stanglmeier¹, S. Königs¹ and M. Zimmermann²

¹ BMW Group, Germany, ² Technical University of Munich, Germany

✉ ferdinand.woehr@bmw.de

Abstract

As current vehicle development processes in the automotive industry are highly distributed, the interaction between design teams is limited. In this paper we use a simulation in order to investigate how the rate of design team interaction affects the solution quality and development cost. Results show, that in case of no limiting constraints, a low rate of interaction yields the best results regarding solution quality and development cost. If design activities are affected by constraints, however, the rate of interaction is subject to a conflict between solution quality and development cost.

Keywords: distributed design, process modelling, process analysis

1. Introduction

Complexity in product development can have different causes. In many cases for instance multiple requirements need to be considered and realized in order to achieve a satisfying product quality. Their contradictory nature often leads to difficulties in component, subsystem and system design. The fulfilment of requirements becomes even more challenging if the functional dependencies between design variables and objective quantities are unknown or undocumented (Toepfer et al., 2018).

Another reason for complexity is the distribution of design activities. The allocation of design variables to different design teams allows the development of large scale subsystems, like a vehicle engine, in decentralized environments. Most of the time, however, this is associated with a limited communication between parties. Hence, highly distributed development processes have to deal with time delays between technical changes in subsystems and the analysis of their impact on the system performance.

An example that contains all of the causes for complexity which are mentioned above is the current vehicle development process. In fact, this process is highly distributed and includes many contradictory requirements. Therefore, vehicle testing and evaluation in early stages of development is crucial. Currently, the integration and verification approach at the system level is similar to a stage-gate-process (Stanglmeier et al., 2018), where information exchange between distributed design teams only happens at certain points in time. Hence, new development processes could provide substantial benefit.

A promising approach for instance is to enable continuous development at the component or subsystem level. This means distributed design teams can seamlessly integrate information about other design variables from different teams into their current work. The impact on solution quality and development cost, however, is difficult to estimate and needs to be quantified. Recent studies on similar subjects have primarily been done by (Xiao et al., 2005) and (Gurnani and Lewis, 2008). They assumed individual

design behaviour to be *rational* or *bounded rational*, which we think does not fully represent the human behaviour in real development processes. Hence, a new model is presented in the context of this study. In this paper we use a simplified process simulation in order to investigate how the rate of design team interaction in distributed development processes affects the solution quality and development cost. Additionally we incorporate possible constraints, like bounds on design variables. Unlike previous investigations we assume gradient-based individual design behaviour, which accounts for the time humans need to find a local solution. As example problem the development of an engine mount system is presented. Results of the study provide quantitative information about the dynamics of distributed development processes with and without constraints that impact the collaboration. This paper is organized as follows: in Section 2 an overview about the current vehicle development process is given and the basic idea of continuous vehicle design is introduced. The simulation model and mathematical formulation of gradient-based individual design behaviour are explained in Section 3. In Section 4 the technical characteristics of the engine mount system are being outlined. Results of the study are presented in Section 5. In Section 6 conclusions are derived and an outlook is presented.

2. Current state of vehicle development

2.1. Methods and processes in early stages

Vehicle development at the system level is responsible for satisfying the requirements that have been assigned to the overall product. Typical requirements are formulated with respect to the aerodynamic drag or structural deformation during a crash accident for example. Common to all of them is that evaluation requires the composition of multiple subsystems, which interact functionally. In order to quantify the structural deformation during a crash accident for example, the deformation of the body, engine and chassis need to be evaluated in combination. In early stages of development these objectives are primarily evaluated by using virtual methods like finite element method (FEM), computational fluid dynamics (CFD) or multi-body simulation (MBS). These methods focus on evaluating quantities of interest from different disciplines and therefore need individual virtual prototypes as input. The vehicle geometry, which serves as an input for the individual prototypes, is assembled every six to ten month. The entire process of prototype assembly, simulation and evaluation can be broken down into 4 steps:

1. *Geometrical data collection*: after component design all parts are collected and assembled to a complete vehicle geometry. This geometry is then stored and cannot be changed afterwards.
2. *Functional model assembly*: based on the geometrical data functional sub models are constructed and combined to virtual prototypes, which serve as an input for the simulation.
3. *Execution of simulations*: within various disciplines simulations are conducted. The generated results provide quantitative information about the current complete vehicle performance.
4. *Evaluation and reporting*: finally, all results are evaluated and presented in different discussion rounds. Optimization potential is transferred to the next iteration of the component design.

A key aspect of this process is the reuse of data. In many cases sub models and simulation results from one discipline need to be integrated in ensuing process steps from other disciplines. As a result, the predefined process is rigid and prohibits the change of component design after data collection. Due to the many unknown dependencies between design variables and objective quantities the effects of design changes can only be identified within the next process iteration.

2.2. The idea of continuous vehicle design

Continuous integration is a process concept mainly used in the software industry in order to develop large and complex software products. There, it already revealed its potential (Stähl and Bosch, 2013). The idea is to enable a flexible integration and verification for components and subsystems based on their design progress or cycle time. As a result, performance testing and evaluation can be done more efficiently and customized to the characteristics of complex systems. The implementation requires a data repository, where defined product elements are stored and preserved. In case a design team decides to release a new version of the element it is responsible for, the corresponding data is submitted to the repository, where automated tests are being conducted. If the new version of the element satisfies all the

requirements, it can be incorporated into the design work of all other design teams. While programming large software products these elements are defined pieces of code, which are developed separately. This process concept can also be applied to the vehicle development. In this case an element would be an engine, wheel or even a screw based on the definition of the process or the decision of the developer. As opposed to a complete vehicle assembly every six to ten month, this approach would enable small integration and verification steps for the individual design teams, depending on their progress, will to synchronize or time schedule. Inevitably an adoption of the process would also increase the design team interaction. However, the effects on solution quality and development cost are unknown. Computational approaches are an appropriate instrument to investigate and quantify these relations.

3. Modelling and simulation of development processes

3.1. Simulation method and parametrized process model

In order to study human collaboration in distributed development processes a simulation model is presented. Unlike empirical techniques computational approaches allow a fast and simple investigation of the problem without changing the real system. A suitable simulation model for the present research objective needs to allow in-depth modelling of human behaviour and information exchange. Especially because distributed development processes are affected by the decisions made in isolation.

Hence, the approach used in this paper focuses on distributed design parties and their time-dependent interaction. The overall framework can be seen as an agent based simulation incorporating game theoretic features. Similar approaches were used by (Lewis and Mistree, 1997) and (Lewis and Mistree, 1998). In the model two parties try to minimize an objective function $f(\mathbf{x})$ (Equation 1). Each of them controls one design variable and has no information about the opposing decision behaviour (Equation 2), which can be modelled as deterministic or stochastic. The framework offers the possibility to incorporate both approaches. Both design variables are also subject to constraints $g_j(\mathbf{x})$ (Equation 3).

$$\min_{x_i} f(\mathbf{x}) \tag{1}$$

$$\mathbf{x} = (x_1, x_2, \dots, x_i) \in \Omega_{ds} \tag{2}$$

$$g_j(\mathbf{x}) \leq 0 \quad (j = 1, \dots, m) \tag{3}$$

As frame of reference we use a parametrized process model (Figure 1). This simplified representation of the vehicle development process allows the investigation of design team interaction. The term design team interaction in general includes many aspects of the human/social encounter. In this paper however, we focus on the exchange of information about the current status of design variables only.

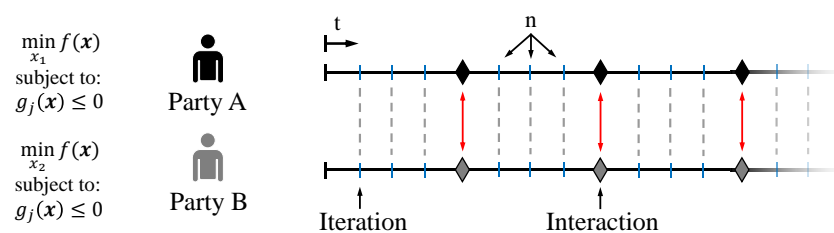


Figure 1. Parametrized process model as frame of reference for the simulation (n=3)

While both parties (A & B) work in parallel they have no information about the decisions made outside of their responsibility. At the beginning of each simulation the initial guess is communicated and serves as an input for the following design steps. As design step, we define every iteration, where both parties have the opportunity to change the value of their design variables. After a certain amount of time, where both parties have worked in isolation, they interact and share the current value of their design variables. We assume that information is only exchanged in constant intervals.

The number of design steps (iterations) between two interactions (n) is an important factor within this study. It describes how many times both parties improve their design without exchanging information. In general, the approach offers two perspectives: parties within the model can either be seen as single

characters or design teams. We assume that in both cases the individual design behaviour can be modelled equally. Hence, insights and findings discovered by the simulation apply to both scenarios.

3.2. State of the art in modelling individual design behaviour

Individual design behaviour in general is associated with many aspects of the human decision making process in engineering. In this paper however, we focus on how single characters or design teams change the value of their design variables over time. In order to demonstrate the difference between various models which have been developed in the past a simple example problem can be used: assume $f(\mathbf{x})$ (Equation 4) represents some quantity of interest that needs to be minimized. The initial guess of both parties is (3, 3). As frame of reference the parametrized process model mentioned above is used. All models are compared after one iteration, thus, no interaction between the parties has taken place.

$$f(\mathbf{x}) = 2x_1^2x_2 + 6x_2^2x_1 \quad (4)$$

In most studies so far the human behaviour has been modelled as *rational* (Devendorf and Lewis, 2011). This means parties derive perfect decisions on their design variables with respect to minimizing their own objective function. Mathematically, this corresponds to the first partial derivative of the assigned objective function set to zero and inserting all the values of the design variables, which are controlled by someone else (Equations 5 and 6). Adopting this model implies that all parties find their optimal solution immediately and without making any mistake. Concerning the example problem mentioned above, parties with a rational design behaviour would change the value of their design variable to (-4.5, -0.5). This model is an example for the deterministic description of the human design behaviour.

$$x_{1,R} = \frac{\partial}{\partial x_1} f(\mathbf{x}) |_{x_2} = 4x_1x_2 + 6x_2^2 = 12x_1 + 54 = 0 \quad (5)$$

$$x_{2,R} = \frac{\partial}{\partial x_2} f(\mathbf{x}) |_{x_1} = 2x_1^2 + 12x_2x_1 = 18 + 36x_2 = 0 \quad (6)$$

More recent studies indicate that modelling individual behaviour as rational does not fully align with the human decision strategy observed in reality (Hey and Orme, 1994). Especially because individual mistakes and human errors are not considered. Hence, (Gurnani and Lewis, 2008) introduced another model, called *bounded rationality*. This mathematical formulation (Equations 7 and 8) includes the possibility of human errors by adding a stochastic term of uncertainty to the equation. Through that, the frequency and significance of mistakes can be modelled. Assuming a standard deviation (σ) of 1, the parties in the example problem, who behave bounded rational, would change the value of their design variable to (-4.3, -0.3). This model combines deterministic and stochastic aspects of human behaviour.

$$x_{1,B} = x_{1,R} + N(0, \sigma^2) \quad (7)$$

$$x_{2,B} = x_{2,R} + N(0, \sigma^2) \quad (8)$$

While the existing models consider the human ability to optimize and cause mistakes, they neglect the time parties need to find a local solution. This can be caused by the limited amount of expert knowledge or constrained computational capability. Either way it leads to a dynamic situation, where the values of the design variables (slowly) converge towards the actual solution. Hence, the individual design behaviour can be compared to a gradient-based optimization algorithm, which uses the slope of a function in order to find a solution (Equations 9 and 10). As a new deterministic model for the individual design behaviour in distributed development processes, we call it *gradient-based behaviour*.

$$x_{1,G}(i+1) = x_{1,G}(i) - \alpha \frac{\partial}{\partial x_1} f(\mathbf{x}) |_{x_{1,G}(i), x_{2,G}(i)} \quad (9)$$

$$x_{2,G}(i+1) = x_{2,G}(i) - \alpha \frac{\partial}{\partial x_2} f(\mathbf{x}) |_{x_{1,G}(i), x_{2,G}(i)} \quad (10)$$

At the beginning the algorithm uses the initial guess of both parties. After that, the values of the design variables are calculated step by step (iteration = i). If no interaction between the parties takes place, each one uses the last known value of the opposing design variable in order to calculate the necessary gradient. In analogy to real design activities the step size (α) can be seen as a measure for the courage of design teams in changing the value of their design variables. Assuming a step size of 0.01, the parties in

the example problem, who behave gradient-based, would change the value of their design variable to (2.1, 1.7). Due to simplicity this paper only focuses on the new deterministic model presented above. Further studies, however, could investigate a combination of deterministic and stochastic models.

3.3. Measuring the process performance

In order to assess the process performance different metrics can be used. Common ones are for instance the product quality or development time. (Stanglmeier et al., 2018) even proposes a method for the financial evaluation of vehicle development processes. In this paper we use two different performance indicators: the solution quality and development cost. The solution quality describes the final value of the objective function, and thus, reveals how good the solution has become, depending on the rate of human collaboration. As development cost we define the effort related to individual iteration and interaction. Therefore, it is rather a theoretical measure and does not provide an accurate prediction of the process cost itself. As a mathematical formulation (Equation 11) this can be written as:

$$Cost = i_{tot}C_i + s_{tot}C_s \quad (11)$$

The meaning of the variables in the equation is shown in Table 1. An example for the cost of an individual iteration could be the time needed in order to conduct a simulation or experiment. On the other hand, cost in terms of interaction could be the effort to share information in a discussion.

Table 1. Variables of the development cost function

Variable	Description
$Cost$	Total development cost
i_{tot}	Total number of iterations
s_{tot}	Total number of interactions
C_i	Cost per iteration
C_s	Cost per interaction

4. Example problem: Design of the engine mount system

4.1. Technical characteristics and multidisciplinary challenges

Automotive engineering is characterized by the consideration of many disciplines, which focus on different vehicle characteristics. This can lead to difficulties in component design because the functional dependencies between disciplines are highly interconnected. An example that represents the challenges of component design in automotive engineering is the development of the engine mount system (Königs and Zimmermann, 2016). In order to provide a connection between the engine-gearbox package and the chassis multiple hydraulic mounts (Figure 2) are placed between both subsystems.



Figure 2. Hydraulic vehicle engine mount (Königs and Zimmermann, 2016)

Their geometrical and functional design (e.g. stiffness, position, orientation) however, has great impact on many disciplines like acoustics, vibrations and durability. In fact, some of the design variables related to the engine mount system even provoke a contrary effect at certain disciplines. A soft rubber core inside of the housing for example is good for the acoustic behaviour, however, it is bad for the durability and vibrations. As a result, conflicts of goals emerge, which increase the challenge of finding a proper solution. Hence, the technical characteristics of the problem itself already induce great difficulties. Furthermore many of the components and subsystems involved are developed by different design teams. While the engine-gearbox package is being managed by the propulsion department, the chassis is in the hands of the vehicle body division. Even the mount system itself is developed by a different design team. Hence, the decentralization and distribution of responsibility induces great difficulties as well.

4.2. Distributed design of the mount position and stiffness

As one of the most important requirements to consider the engine mount system needs to avoid the transmission of vibrations, which are initially caused by the combustion process inside of the powertrain. In case of a bad component design these vibrations are transmitted to the passenger compartment, where they can cause audible noise. In order to avoid these effects, it is necessary to decouple the rotational eigenfrequency of the engine-gearbox package f_{Eig} from the ignition frequency. As the operational conditions of the engine are usually not considered as design variables the goal is to minimize the rotational eigenfrequency as much as possible. This can be done by changing the values of certain design variables related to the engine mount system. Assuming a simplified dependency between the acoustic behaviour and design variables the mathematical formulation of the problem is similar to Equation (13):

$$f_{Eig}(\mathbf{x}) = \sqrt{\frac{y_{pos}^2 c_{dyn} + z_{pos}^2 \frac{c_{dyn}}{z}}{I_{xx}}} \quad (13)$$

Hence, the rotational eigenfrequency depends on three different types of design variables. First, the location of the mount can be used in order to reduce the acoustic mode. By minimizing the distance between the torque-roll axis and the mount position (y_{pos}, z_{pos}) the tendency of a high natural oscillation is decreased. Another factor of influence is the dynamic stiffness (c_{dyn}) . By definition the dynamic stiffness corresponds to the mechanical behaviour of the rubber core under deformation at a certain frequency. Finally, the moment of inertia of the engine-gearbox package (I_{xx}) has an impact as well. The moment of inertia depends on the shape and mass distribution of the engine-gearbox package and its attached surroundings. In this paper we assume that the y-position (y_{pos}) and dynamic stiffness (c_{dyn}) of the engine mount system are being controlled by two different parties (Figure 3). They both work in parallel and try to minimize the rotational eigenfrequency. All other design variables are constant and not allocated.

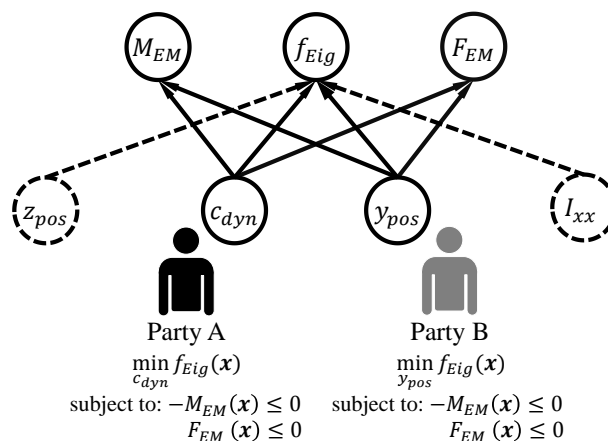


Figure 3. Distributed development of the mount position and stiffness

However, the y-position and dynamic stiffness of the mount system can not only be optimized with respect to the rotational eigenfrequency, but have to be designed under the consideration of certain constraints as

well. A major restriction for instance comes from the maximum engine torque leading to a minimal moment that the mount system has to resist, without leaving the linear region of its characteristic force-deformation curve. A functional approximation for that moment (M_{EM}) is shown in Equation (14):

$$M_{EM} \sim c_{dyn} y_{pos} \quad (14)$$

Another restriction is the maximum force (F_{EM}), which applies during the engine start and shall not exceed a critical value. In case of a bad component design the chassis is oscillating with an unacceptable decay rate. An approximation of the functional dependency is shown in Equation (15):

$$F_{EM} \sim c_{dyn} - y_{pos} \quad (15)$$

In order to study the effects of critical constraints on human collaboration two scenarios were investigated. In a first test series (scenario 1) the design space of both parties was limited by a constraint, that ensures that the moment restriction is satisfied (Figure 4 - left side). This corresponds to a development process, where the requirements do not impact the collaboration. In a second test series (scenario 2) the design space of both parties was additionally limited by a constraint, that ensures that the force restriction is satisfied (Figure 4 - right side). This is supposed to represent a development process, where the requirements impact the collaboration considerably.

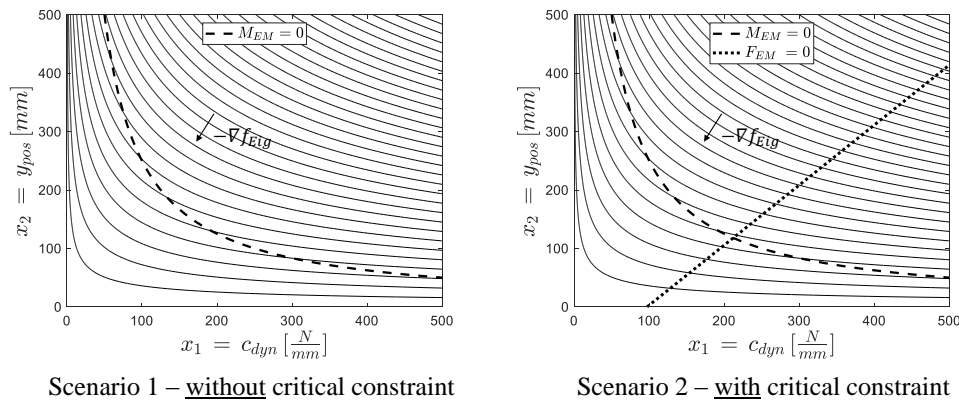


Figure 4. Solution of the rotational eigenfrequency in scenario 1 and scenario 2

A constraint defines an area, where no satisfying solution can be found. Hence, reaching the border between a permissible space and constrained area requires a human decision with respect to accepting or rejecting the value of a design variable. We assume that parties reject values of their design variables, which lie in a constrained area. If a constraint impacts the collaboration, we called it critical.

4.3. Configuration of the simulation study

After introducing all relevant information the simulation model (section 3) is applied to the example problem (section 4). This allows a computational analysis of the present research objective. However, due to the simplification of the process model and engine mount system, the study only provides basic information about distributed development processes without the intend to fully represent reality.

In total eight simulations have been carried out with a total number of twenty iterations each. While in both scenarios the same four simulations have been conducted, the rate of interaction (n) was varied between one, five, ten and twenty. Individual design behaviour has been modelled as gradient-based.

5. Results and discussion

5.1. Time-dependent evolution of design variables

First, the evolution of the design variables is being discussed in order to understand the basic effects related to distributed design (Figure 5). While the left side represents the consideration of one constraint (scenario 1), the right side displays the results incorporating two constraints (scenario 2). The markers represent the following: ▼ indicates the initial guess ($c_{dyn} = 450 \text{ N/mm}$ and $y_{pos} = 450 \text{ mm}$) of both parties. These

values are typical for a predecessor vehicle, which is normally used as reference at the start of the development process ● represents the design choices of party A. ■ represents the design choices of party B. ▲ illustrates the interaction. The step size of both parties is 160, which can be seen as a realistic value compared to the individual design behaviour observed in reality.

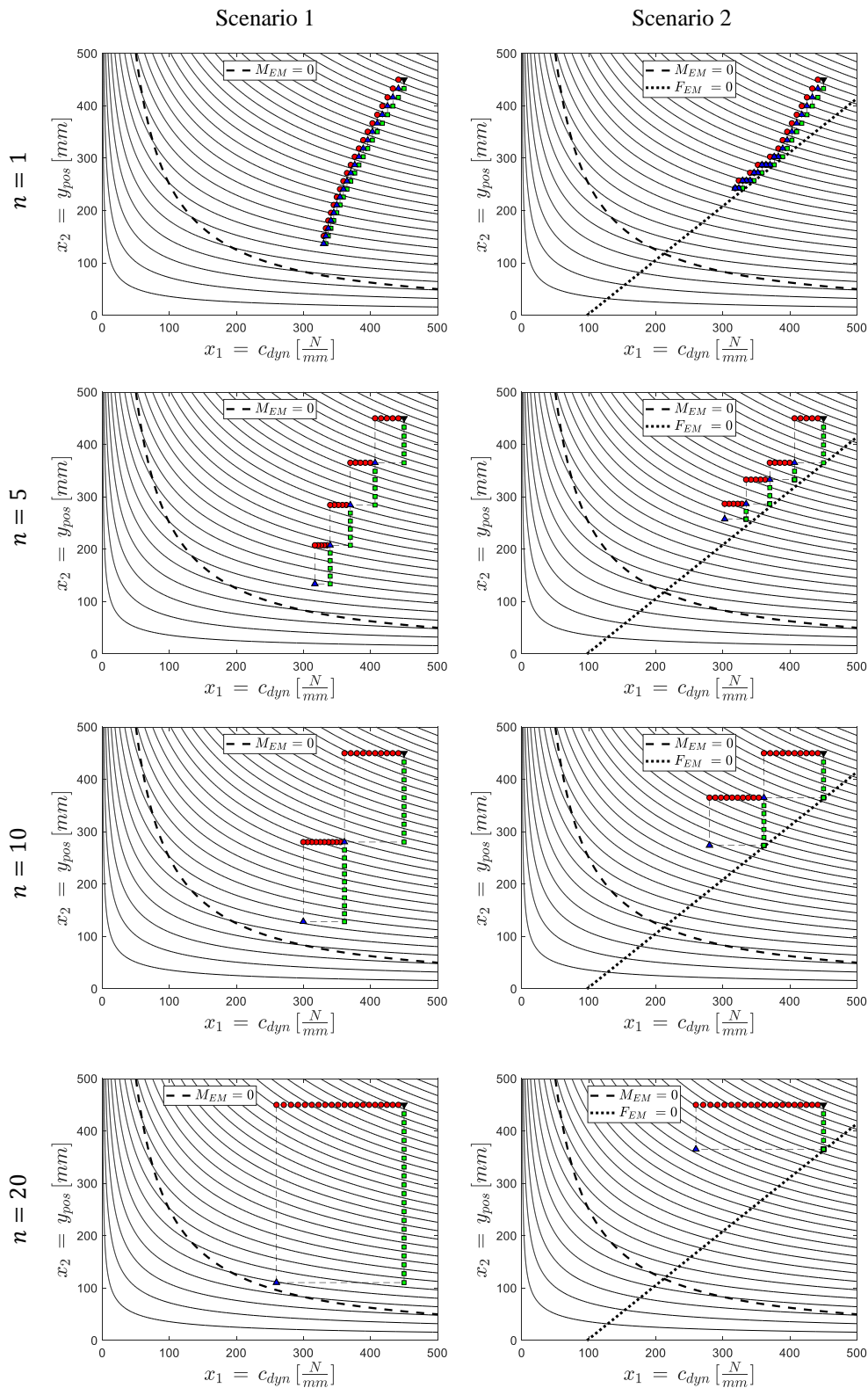


Figure 5. Evolution of design variables over time for scenario 1 (left) and scenario 2 (right)

In scenario 1 the time-dependent evolution of the design variables is not affected by the constraint, because the border of the restricted area is not reached. As a result, the solution quality only depends on the number of design team interactions. In case of a high rate of interaction ($n = 1$) the design choices of both parties only diverge slightly before information about the current status is shared and the next iteration takes place. Hence, the sum of all design activities (iterations and interactions) equals a smooth curve directed towards the solution space. On the other hand, a low rate of interaction ($n = 20$) leads to a highly diverging behaviour of the design choices between each information exchange. This may cause reaching the border of a constrained area without realising it, because both parties work with an obsolete status of the other design variable. In this particular case, however, less information exchange between the both parties (A & B) leads to a higher solution quality.

In scenario 2 the time-dependent evolution of the design variables is highly affected by the additional constraint, because the restricted area is reached by party B after a certain number of iterations. Hence, the solution quality at the end of the simulation depends on the design team interaction and the characteristics of the constraint. Especially if the information exchange between both parties is rare ($n=20$), the number of iterations, where party B cannot further minimize, is high. In case of a high rate of interaction ($n=1$) however, party B only needs to wait a short time until the most recent status of party A is shared, and thus, a new minimization step is possible. Hence, more information exchange and in particular at the right time reduces the number of iterations, where party B is locked into an unfavourable position and cannot further optimize. In general, scenario 2 reveals that more information exchange between the both parties leads to a higher solution quality.

In summary, this first analysis without the consideration of cost and effort provides two insights. In case of no critical constraint, less interaction between both parties leads to a higher solution quality. However, if the design activities are affected by a constraint, more information exchange between the both parties leads to a higher solution quality. All insights only apply to this specific dependency function.

5.2. Evaluation of solution quality vs. development cost

As part of a second analysis all of the eight simulations which have been conducted were evaluated regarding development cost. In this context, the cost per iteration and interaction was assumed to be 1. Hence, every simulation can be described by two characteristic values: the solution quality and development cost. Depending on the collaboration and scenario both values are shown in Figure 6. While a smaller rotational eigenfrequency and lower development cost are assumed to be better, the axis have been labelled additionally.

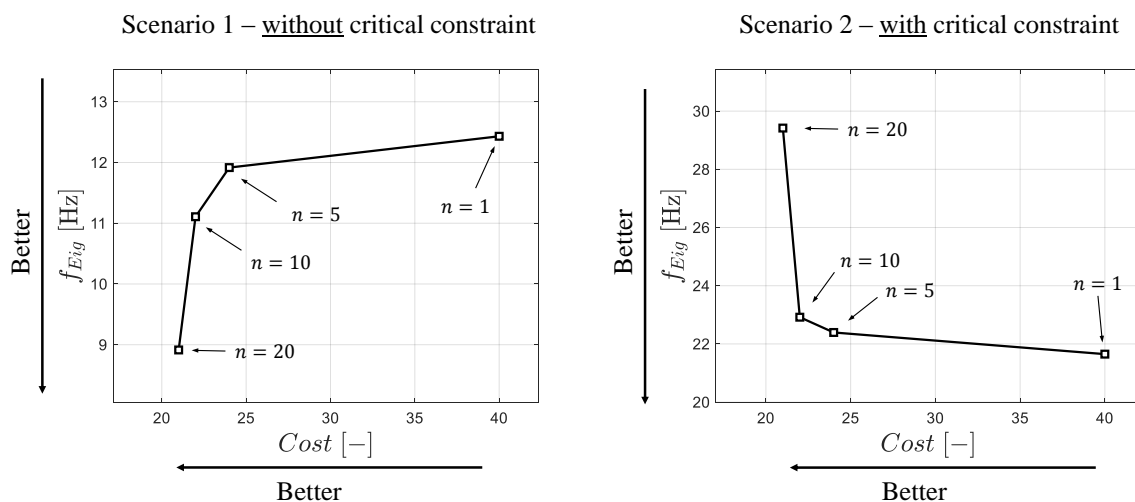


Figure 6. Solution quality vs. development cost for scenario 1 (left) and scenario 2 (right)

In scenario 1 less interaction between the parties leads to a higher solution quality and lower development cost. However, the benefits of limited collaboration need to be evaluated in detail. While between a high ($n=1$) and an intermediate ($n=5$) rate of interaction the advantage is primarily caused by

the reduction of cost, the gain between an intermediate and small rate of interaction ($n=20$) is due to the solution quality increase. In any case, it makes sense to decrease the information exchange between the parties in order to obtain a satisfying result without spending too much unnecessary effort.

In scenario 2 less interaction between the parties leads to a different behaviour of the solution quality and development cost. While the limitation of collaboration decreases the development cost, the solution quality is lowered as well. Here again, the effects need to be evaluated in detail. By changing the rate of interaction from ($n=1$) to ($n=5$) the cost can be reduced significantly, while the solution quality only drops slightly. A further limitation of the information exchange, however, reduces the solution quality significantly while the cost only drops slightly. Hence, reducing the interaction from a high to an intermediate rate proves to be more efficient than limiting the collaboration completely.

In summary, this second analysis provides two insights. In case of no critical constraint, a low rate of interaction yields the best results regarding the solution quality and development cost. Hence, no conflict between the both performance indicators occurs. In case the design activities are affected by a constraint, however, more information exchange between the parties leads to a higher solution quality but also increases the development cost. Hence, a conflict between the both performance indicators is present.

6. Conclusion and outlook

In this paper a simplified process simulation was used in order to study human collaboration in distributed development processes. By varying the amount of information exchange between parties the following insights were produced: if design activities are not affected by constraints, a low rate of interaction leads to the best results regarding solution quality and development cost. In the presence of critical constraints, however, the rate of design team interaction is subject to a conflict between solution quality and development cost. Since the results depend on the example problem and its functional characteristics the findings are not applicable in general.

Further studies will provide an extension of the parametrized process model. In order to investigate real development processes additional features will be incorporated, which account for multiple parties, asymmetric information exchange and irregular iteration. Hence, more complex development processes can be studied. Another focus will lie on the empirical validation of the theoretical simulation results.

References

- Devendorf, E. and Lewis, K. (2011), "The Impact of Process Architecture on Equilibrium Stability in Distributed Design", *Journal of Mechanical Design*, Vol. 133 No. 10. <https://doi.org/10.1115/1.4004463>
- Gurnani, A. and Lewis, K. (2008), "Collaborative, Decentralized Engineering Design at the Edge of Rationality", *Journal of Mechanical Design*, Vol. 130 No. 12. <https://doi.org/10.1115/1.2988479>
- Hey, J.D. and Orme, C. (1994), "Investigating Generalizations of Expected Utility Theory Using Experimental Data", *Econometrica: Journal of the Econometric Society*, Vol. 62 No. 6. <https://doi.org/10.2307/2951750>
- Königs, S. and Zimmermann M. (2016), "Resolving Conflicts of Goals in Complex Design Processes - Application to the Design of Engine Mount Systems", *Proceedings of the 7th International Munich Chassis Symposium*. https://doi.org/10.1007/978-3-658-14219-3_14
- Lewis, K. and Mistree, F. (1998), "Collaborative, Sequential and Isolated Decisions in Design", *Journal of Mechanical Design*, Vol. 120 No. 4. <https://doi.org/10.1115/1.2829327>
- Lewis, K. and Mistree, F. (1997), "Modelling Interactions in Multidisciplinary Design: A Game Theoretic Approach", *AIAA Journal*, Vol. 35 No. 8. <https://doi.org/10.2514/2.248>
- Stähl, D. and Bosch, J. (2013), "Experienced benefits of continuous integration in industry software product development: A case study", *Proceedings of the IASTED International Conference on Software Engineering*. <https://doi.org/10.2316/P.2013.796-012>
- Stanglmeier, M. et al. (2018), "Economic assessment of virtual validation processes in the automotive development - An analytical approach", *Proceedings of the 2018 Annual IEEE International Systems Conference (SysCon)*. <https://doi.org/10.1109/SYSCON.2018.8369496>
- Toepfer, F. et al. (2018), "Integrating the knowledge about functional interdependencies into a parameter management approach", *Proceedings of the DESIGN 2018 15th International Design Conference*. <https://doi.org/10.21278/idc.2018.0222>
- Xiao, A. et al. (2005), "Collaborative Multidisciplinary Decision Making Using Game Theory and Design Capability Indices", *Research in Engineering Design*, Vol. 16 No. 1-2. <https://doi.org/10.1007/s00163-005-0007-x>