KEYWORDS Project dynamics Cooperative work Work transformation matrix

ESTIMATION OF WORK TRANSFORMATION MATRICES FOR LARGE-SCALE CONCURRENT ENGINEERING PROJECTS

Christopher M. Schlick [1]
[1] Institute of Industrial Engineering and Ergonomics, RWTH Aachen University, Germany
c.schlick@iaw.rwth-aachen.de

Sebastian Terstegen [1]
s.terstegen@iaw.rwth-aachen.de

Sönke Duckwitz [1]
s.duckwitz@iaw.rwth-aachen.de

ABSTRACT

This paper presents an approach to estimating work transformation matrices of large-scale concurrent engineering projects with periodically correlated work processes. A mathematical model of cooperative work is formulated on the basis of the theory of stochastic periodic autoregressive processes. The model can capture not only the cooperative processing of the development tasks with short iteration length but also long-scale effects of withholding the release of information on purpose. This strategy aims at improving the implementation of the product architecture. The solutions of the least-square estimates of the work transformation matrices and the covariances are given. Finally, the estimation accuracy is investigated based on a simulated development of a door module of a vehicle door.

1. Introduction

Effective planning, scheduling and control are essential in large-scale concurrent engineering *(CE)* projects. In these projects a systematic approach to the integrated, concurrent design of the product and the related manufacturing processes is taken to cause the developers, from the outset, to consider all elements of the product life cycle. The CE teams are often structured according to the product architecture and are coordinated by system-integration and management teams. Depending on the intensity of the cooperative relationships, some teams can enter cycles of multiple revisions, which can lead to long delays. Moreover, a fatal pattern termed "design churns" *(Yassine et al., 2003)* or "problem-solving oscillations" *(Mihm and Loch, 2006)* can emerge. In this case, the project irregularly oscillates between being on, ahead of, or behind schedule. A deeper understanding of dynamical dependency structures together with novel management methods are needed to successfully plan, schedule and control these kinds of projects. The goal of this paper is therefore twofold: first, to present a model of cooperative work in large-scale CE projects that is based on Work Transformation Matrices *(Smith and Eppinger, 1997)* as a task-oriented variant of the Design Structure Matrix; second, to introduce an accurate and efficient procedure for estimating the free parameters from field data.

2. Models of Cooperative Work in Large Scale NPD Projects

To analyze the design process, models of cooperative work with different strength were formulated. According to our previous work *(e.g. Schlick et al., 2013)*, distinct phases in CE projects with p fully concurrent tasks can be represented by the recurrent relation:

$$X_t = A_0 \cdot X_{t-1} + \varepsilon_t \quad t \geq 1. \quad (1)$$

The random variable $X_t \epsilon [0;1]^p$ represents the work remaining at time step t. The work remaining can be measured by the labor units that are required to complete a particular development activity. $A_0 = (a_{ij})$ is the $p \times p$ Work Transformation Matrix *(WTM)*. We use the improved WTM concept of Yassine et al. *(2003)* and Huberman and Wilkinson *(2005)* in which the diagonal elements a_{ii} denote the autonomous task processing rates. This is in contrast to the original WTM model of Smith and Eppinger *(1997)* in which tasks are processed at the same rate. We assume that all tasks are initially 100% to be completed, and so the initial state is $x_0 = (1 \quad 1 \quad \cdots \quad 1)^{\mathrm{T}}$. The zero-mean Gaussian random variable $\varepsilon_t = N(x; 0, C)$ is added to model unpredictable performance fluctuations.

A logical extension is to formulate a periodic vector autoregressive *(PVAR)* model *(Ursu and Duchesne, 2009)*, which can capture not only the cooperative processing of the development tasks with short iteration length but also the long-scale effect of inadvertent information hiding. Short iterations are necessary to process and disseminate component-related information within CE teams. However, the product architecture must also be taken into account and the various components must be integrated to work efficiently together *(Eppinger and Browning, 2012)*. The usual organization of the work is such that the tasks are cooperatively processed in the sense that information about functional or topological entities is widely disseminated within and between teams, not hidden. A long-scale effect, however, can also occur, because teams focusing on system-level design may withhold the release of information on purpose *(Yassine et al. 2003)*. Such a hold-and-release policy is typical for CE projects in the automotive industry. This behavior aims at improving the implementation of the product architecture and releasing only system designs that have a certain level of maturity. A corresponding deterministic model was developed by Yassine et al. *(2003)* and we built directly upon their results. However, our approach can also account for performance fluctuations and the free parameters can be estimated accurately by ordinary least squares. For the sake of simplicity, we model a hierarchical coordination structure with only two levels, namely system-level and components-level design. We assume that a fraction of finished work is released by the teams responsible for system integration and testing to component-level design teams only at time steps *ns* ($n \geq 1$, $s \geq 2$). At all other time steps the tasks are processed by short iterations without withholding information. State eq. 1 can be generalized to a process with periodically correlated components:

$$X_{ns+v} = \Phi_1(v) \cdot X_{ns+v-1} + \varepsilon_{ns+v}, \quad (2)$$

where the index n indicates the long time scale with period s, and v the short time scale. $X_{ns+v} = (X_{ns+v}(1) \quad \cdots \quad X_{ns+v}(d))^{\top}$ is a random vector. The leading vector components represent the work remaining related to component-level and system-level design tasks being processed on the short time scale. Their processing can be modeled by submatrices A_0^C, A_0^S, A_0^{SC} and A_0^{CS}. A_0^C represents the processing of component-level tasks. A_0^S refers to system-level tasks in an analogous manner. A_0^{SC} determines the fraction of labor units generated by system-level design for the component-level tasks, whereas A_0^{CS} determines the fraction generated from component-level to system-level. Moreover, the states have to be augmented to store the amount of finished work on the system level that is accumulated over the short iterations. The finished work remains hidden for the component-level teams until it is released at time step *ns*. The submatrix A_0^{SH} captures the fraction of labor units generated by the system-level tasks in each iteration at time steps $v=1, \ldots, s-1$. After release, additional work is generated for the component-level tasks. This work is calculated based on A_0^{HC}. The states of the concurrent work processes are expressed by X_{ns+v} based on the autoregressive coefficients $\Phi_1(v)$. The two time scales correspond to indices n and v. To define the coefficients in state eq. 2, two WTMs $\Phi_1(s)$ and $\Phi_1(1)$ are introduced. The release of hidden information at time steps *ns* ($n=1, 2, \ldots$) is modeled by $\Phi_1(s)$:

$$\Phi_1(s) = \begin{pmatrix} A_0^C & A_0^{SC} & A_0^{HC} \\ A_0^{CS} & A_0^S & 0 \\ 0 & 0 & \{\varepsilon\} \cdot I \end{pmatrix} \quad (3)$$

ε denotes an arbitrarily small positive quantity, which is necessary to explicitly evaluate emergent complexity *(see Schlick et al., 2011)*. I denotes an identity matrix. The size of I corresponds to the number system-level tasks that are processed on the short time scale. The task processing in the iterations before release is modeled on the basis of $\Phi_1(v)(v = 1, \ldots, s-1)$. It is assumed that the coefficients are constant during one period, and it holds that:

$$\Phi_1(1) = \cdots = \Phi_1(s-1) = \begin{pmatrix} A_0^C & A_0^{SC} & 0 \\ A_0^{CS} & A_0^S & 0 \\ 0 & A_0^{SH} & \{1-\varepsilon\} \cdot I \end{pmatrix} \quad (4)$$

ε_{ns+v} corresponds to a zero-mean periodic white noise.

3. Least-Squares Estimation of Work Transformation Matrices

The free parameters of a PVAR model without linear constraints can be calculated efficiently based on least-squares estimation techniques from textbooks *(see, e.g. Lütkepohl, 2005)*. However, in the developed model formulation we had to pose the constraint that some entries of $\Phi_1(1)$ and $\Phi_1(s)$ must be zero so that the hold-and-release policy can be modeled. Furthermore, some coefficients are linear dependent. We therefore cannot use standard techniques. Instead, we can apply a method developed by Ursu and Duchesne *(2009)* for least-squares estimation with linear constraints on the regression parameters. For a convenient state representation let

$$\begin{aligned} Z(v) &= \left(X_v, \ldots, X_{(T-1)s+v}\right) \\ E(v) &= \left(\varepsilon_v, \ldots, \varepsilon_{(T-1)s+v}\right) \quad (5) \\ \boldsymbol{X}(v) &= \left(X_{v-1}, \ldots, X_{(T-1)s+v-1}\right) \end{aligned}$$

denote the random vectors of work remaining of the component-level and system-level design tasks. Since for all short iterations $v=1, \ldots, s-1$ the regression equation contains the same unknown regression parameters, we can concatenate them:

$$Z_c = \Phi_1(1) \cdot X_c + E_c, \text{ where} \quad (6)$$

$$\begin{aligned} Z_c &= \left(Z(1), \ldots, Z(s-1)\right) \\ \boldsymbol{X}_c &= \left(\boldsymbol{X}(1), \ldots, \boldsymbol{X}(s-1)\right) \\ E_c &= \left(E(1), \ldots, E(s-1)\right). \quad (7) \end{aligned}$$

The regression coefficients are given by $\beta(v) = vec[\Phi_1(v)]$. The combined vector of regression coefficients

$$\beta = \begin{pmatrix} \beta(1) \\ \beta(s) \end{pmatrix} \quad (8)$$

contains in parts the same elements i.e. they are linear dependent. Furthermore, many elements of β are known to be zero (cf. eqs. 3 and 4). This can be expressed by the linear relation

$$\beta = R \cdot \xi + b. \quad (9)$$

ξ represents the vector of unknown regression parameters. R is the matrix of constraints on components of ξ. b expresses the values of the constraints. When we convert the above linear relation into a vector representation we arrive at:

$$z = (X^{\top} \otimes I_d) \cdot \beta + e = (X^{\top} \otimes I_d) \cdot (R \cdot \xi + b) + e, \text{ where} \quad (10)$$

$$X^{\top} = \begin{pmatrix} X_c^{\top} & 0 \\ 0 & X^{\top}(s) \end{pmatrix}. \quad (11)$$

I_d denotes an identity matrix of size d. The combined noise vector is given by

$$e = \begin{pmatrix} vec[E_c] \\ vec[E(s)] \end{pmatrix}. \quad (12)$$

According to Ursu and Duchesne *(2009)* the least-squares estimator of ξ under the formulated linear constraints is given by

$$\hat{\xi} = \left(R^{\top}\left(X \cdot X^{\top} \otimes \tilde{C}_e^{-1}\right)R\right)^{-1} R^{\top}\left(X \otimes \tilde{C}_e^{-1}\right)\left(z - (X^{\top} \otimes I_d)b\right) \quad (13)$$

and a consistent estimator of the covariance C_e is given by

$$\tilde{C}_e = \left\{\frac{1}{T-d}\right\} \cdot \left(Z - \hat{B} \cdot X\right)\left(Z - \hat{B} \cdot X\right)^{\top} \quad (14)$$

$\hat{B}$ denotes the least-squares estimate for the full non constraint case that is based on

$$\hat{B}(v) = Z(v) \cdot \boldsymbol{X}^{\top}(v)\left(\boldsymbol{X}(v) \cdot \boldsymbol{X}^{\top}(v)\right)^{-1}. \quad (15)$$

The resulting estimator of β is given by $\hat{\beta} = R \cdot \hat{\xi} + b$.

4. Validation Study

To evaluate the accuracy of the estimation technique in a setting with complete control of confounding factors, we carried out a simulation study. We formulated a PVAR model that connects the dynamics of module design and integration in a vehicle door development project with component development. This model forms a reference model that was used to simulate task processing and generate time series of work remaining of different length. We then tried to "reconstruct" the reference model representation purely from data. Technically speaking, we investigate the identifiability of the reference model and the associated parameter uncertainty. The main question is how the estimation accuracy of model matrices is influenced by the number of data points T that are available for numerical estimation. To simplify the analysis, we focus on the door module of the vehicle door subsystem. A door module typically consists of a functional carrier plate and other components that are fitted on it. Various door components, such as the window lift mechanism, locks, wiring harness, switches, loud speakers, crash sensors and cables connecting the latch to the inner release handle have to be integrated. Design, development and manufacturing of the door module is typically outsourced from the major automotive manufacturers to selected first-tier suppliers to save costs and weight. We focused on the periodically correlated work processes in the development organization of the supplier. At the supplier, the corresponding module development project is usually structured on the first level of the work breakdown structure according to the main project functions such as engineering design, manufacturing, procurement, sales, controlling, quality management etc. To build the reference model we put the engineering design and the related component development and integration activities into focus. We had to greatly simplify the real project organization to obtain simulation models of reasonable complexity. However, this does not limit the generality of the model-driven approach. We assume that the project-related work in the engineering design department of the first-tier supplier mirrors the system structure of the door module and breaks it down into development teams focusing on either mechanical or electrical/electronic functions. The systems engineering process and the integration of the components into a fully functional module is carried out by a dedicated module design and integration testing team. This team also coordinates the design of the interfaces to the complete door system and the car body. We focus on the cooperation between the module design and integration testing team and two subordinate teams dealing with the mechanical design of the functional carrier plate and the mechanical/kinematic design of the window lift mechanism. According to section 2, information about integration and tests of specific geometric/kinematic entities is inadvertently hidden by the module design team for a certain fraction of time and not immediately disseminated to component-level teams. To simplify the evaluation of the accuracy of the estimation methods, we define a PVAR model that includes only one module-level task and two component-level tasks. The three different teams process the tasks simultaneously. Each team is assigned one task in the model. The three vector components of the state variable X_{ns+v} represent the relative number of issues that need to be resolved before final design release. We assume that the first component-level team, which designs the functional carrier plate, works with autonomous task processing rate $a_{11}^{C}=0.91$. The second component-level team, which designs the window lift mechanisms, can use a standard mechanism and several internally standardized parts and therefore processes the task faster, with autonomous task processing rate $a_{22}^{C}=0.88$. The cooperative relationships within both component-level teams are very similar and therefore the tasks are coupled with symmetric strength $a_{12}^{C}=a_{21}^{C}=0.03$. Due to a well-designed system architecture, the team responsible for module design and integration testing works with autonomous task processing rate $a^{S}=0.80$ (please note that for reasons of internal consistency we have used the superscript "S" and not "M" to model the task processing on the module level). The module-level task generates 5% of finished work at each short iteration that is put in a hold state until it is released at time step ns. Hence, $a^{SH}=0.05$. Furthermore, both component-level teams generate 5% of finished work at each iteration for the module-level, and we have $a_{11}^{CS}=a_{12}^{CS}=0.05$. Conversely, the module design team only feeds back 2% of the unresolved issues at each short iteration, and there is $a_{11}^{SC}=a_{21}^{SC}= 0.02$. The accumulated issues of the module-level are released to the first component-level team at the end of the period ($a_{11}^{HC}=1$ and $a_{21}^{HC}=0$). This team transfers the accumulated unresolved issues to the second component-level team at the next time step. Additional dynamical dependencies were not considered and therefore all other matrix entries were defined as zero. We calculated with $\varepsilon=0.001$. The complete representation related to the autoregressive coefficients is:

$$A_0^C = \begin{pmatrix} 0.91 & 0.03 \\ 0.03 & 0.88 \end{pmatrix} \quad (16)$$

$$A_0^S = a^S = 0.80 \quad (17)$$

$$A_0^{CS} = (0.05 \quad 0.05) \quad (18)$$

$$A_0^{SC} = \begin{pmatrix} 0.02 \\ 0.02 \end{pmatrix} \quad (19)$$

$$A_0^{SH} = a^{SH} = 0.05 \quad (20)$$

$$A_0^{HC} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad (21)$$

$$\Phi_1(s) = \begin{pmatrix} 0.91 & 0.03 & 0.02 & 0 \\ 0.03 & 0.88 & 0.02 & 0 \\ 0.05 & 0.05 & 0.8 & 0 \\ 0 & 0 & 0.05 & 0.9999 \end{pmatrix} \quad (22)$$

$$\Phi_1(1) = \begin{pmatrix} 0.91 & 0.03 & 0.02 & 1 \\ 0.03 & 0.88 & 0.02 & 0 \\ 0.05 & 0.05 & 0.8 & 0 \\ 0 & 0 & 0 & 0.0001 \end{pmatrix}. \quad (23)$$

The initial state was defined using the assumption that all development tasks are to be fully completed initially and that no issues are left unresolved from the previous ideation phase. Furthermore, we have to make reasonable assumptions about the variances and covariances of the unpredictable performance fluctuations which are represented by the random variable ε_{ns+v}. We assumed that the standard deviation σ_{ii} of performance fluctuations influencing task i in the module development project is proportional to the autonomous task processing rate: as the task processing rate increases *(i.e. as the speed of task processing slows down)*, so the expected root square deviation from the mean will also increase. We chose a proportionality constant of $r=0.02$. Additional correlations among vector components were not considered. We also assumed that the variance of the fluctuations related to the issues put in a hold state is reduced by the factor 10^{-3} and that the same reduced variance holds for the fluctuations when releasing the hidden information. Through these variance reductions, the performance variability related to the augmented substate accounting for the periodic hold-and-release policy is extremely small and does not influence the basic mechanisms of cooperation between teams in the model. Due to space limitations, we do not show the variances and covariances. However, to get an impression of the dynamics of task processing, Figure 1 shows the results of a typical run of the Monte Carlo simulation for a release period of $s=4$ [weeks].

The finished work that was put in hold state by the team responsible for module design and integration testing at each short iteration is also shown in the list plot of **Figure 1** around the abcissa.

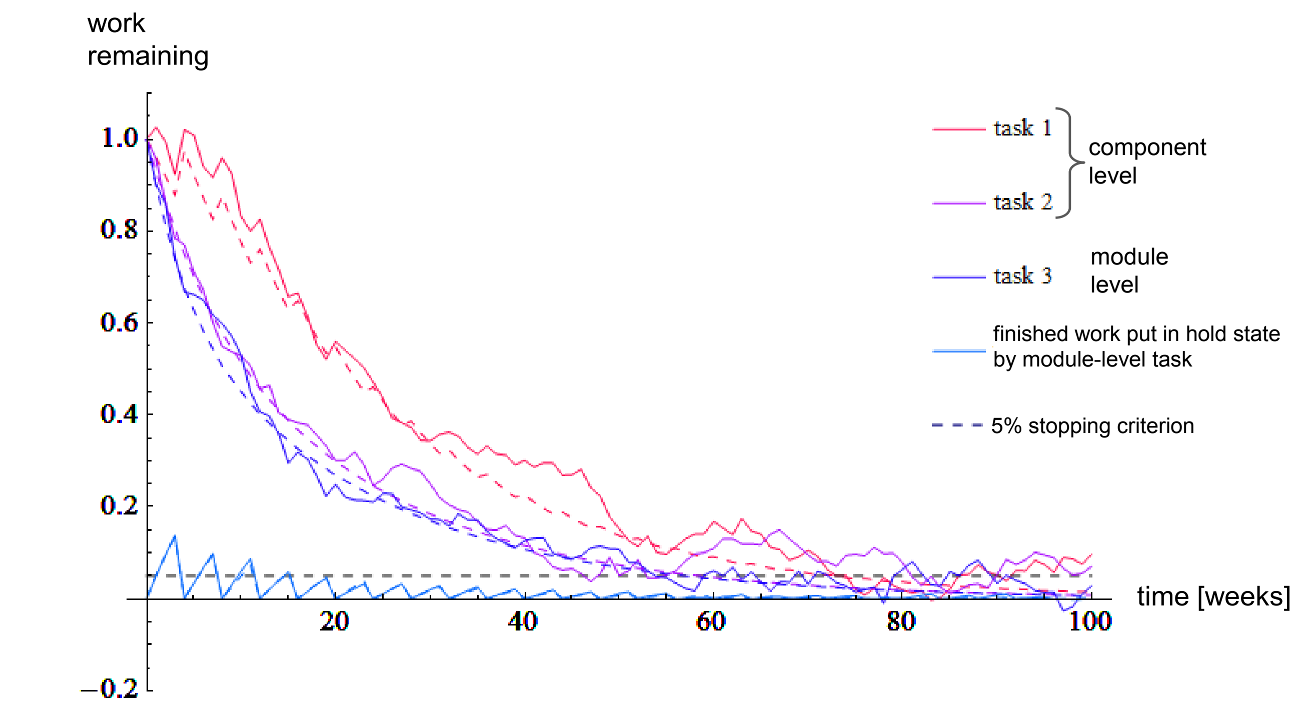

FIGURE 1. *List plot of work remaining in simulated door module development with correlated work processes. The processing of three development tasks is shown. The release period is s=4 [weeks]. The data is based on a single run of the Monte Carlo experiment. The parameters are given by eqs. 16 to 23. The plot also shows the means of task processing as dashed curves. The stopping criterion of 5% is marked by a dashed line at the bottom.*

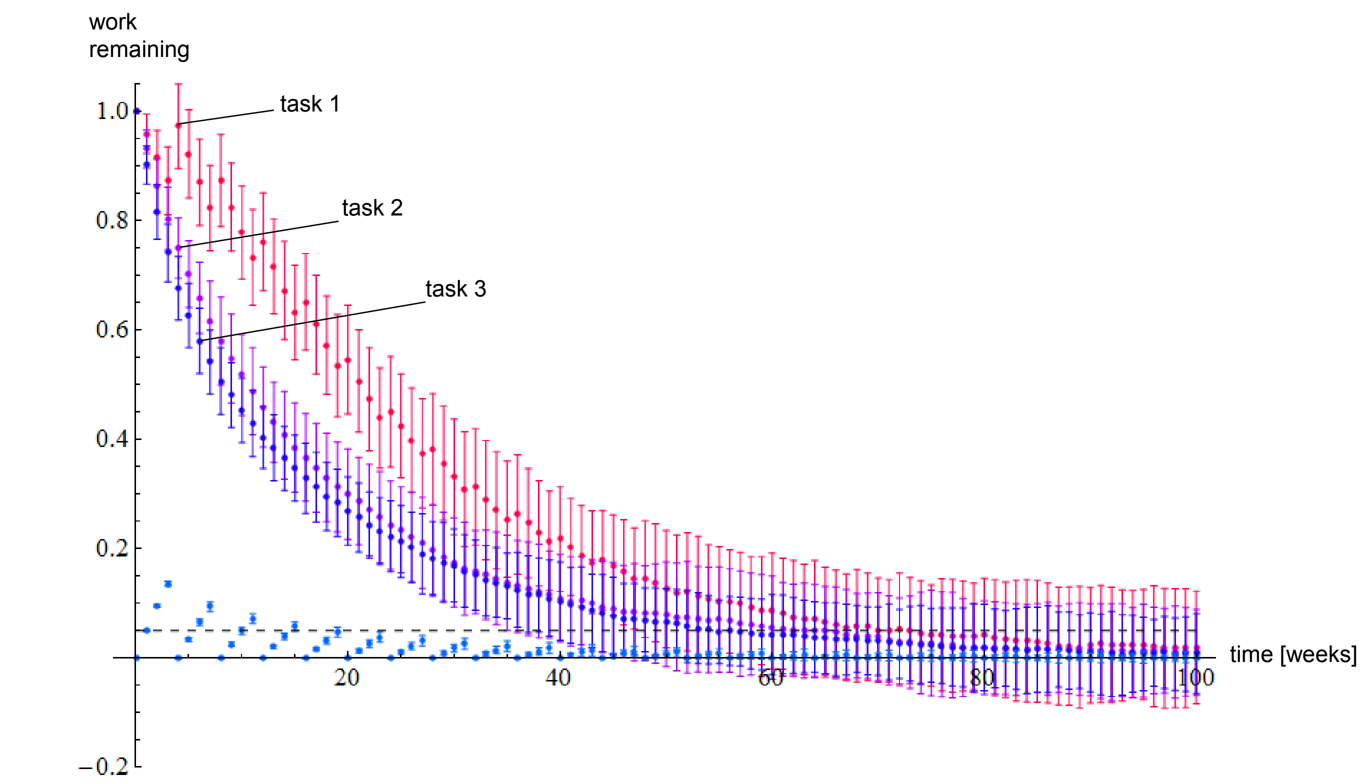

FIGURE 2. *Error list plot of work remaining in simulated door module development with correlated work processes. The release period is s=4 [weeks]. A total of 100 separate and independent runs were calculated. The plot shows means of work remaining as note points and 95% confidence intervals as error bars. The simulation conditions and parameters are the same as in Fig. 1.*

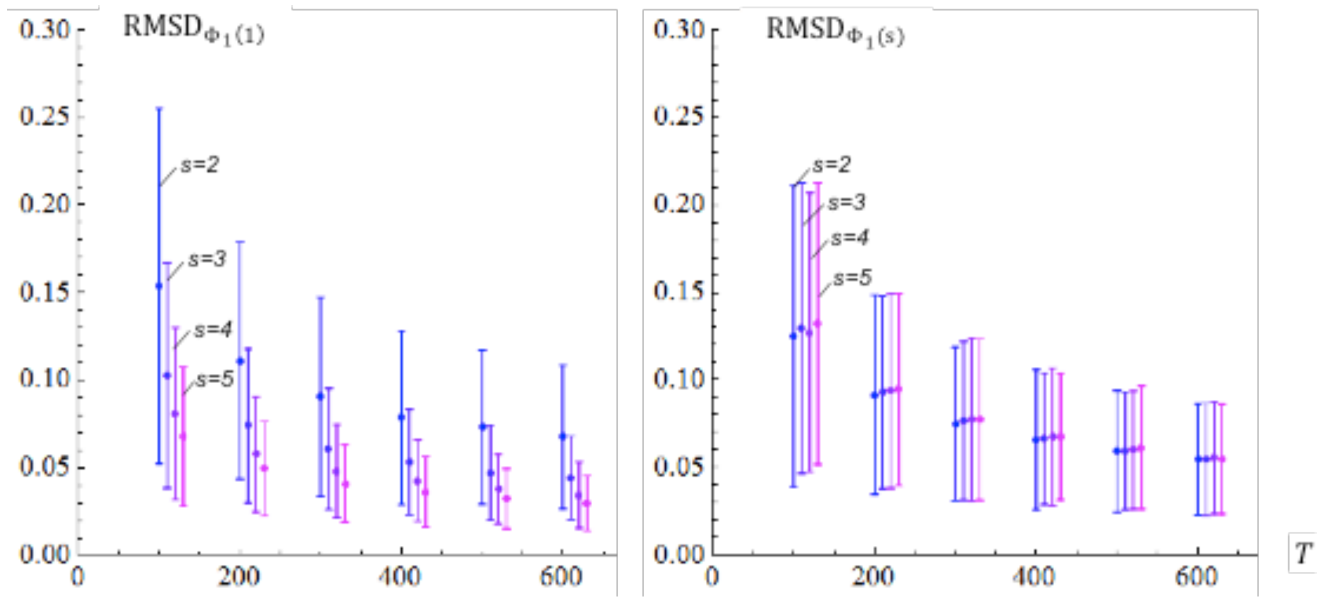

FIGURE 3. *Root mean square deviation (RMSD) between true model matrices and their least-squares estimates. The release period s of finished work put in hold state at each short iteration was varied between 2 and 5 [weeks]. The number of data points T used for estimation was varied within 100 and 600 time steps. 1000 independent runs were calculated.*

Seeing as it is not meaningful to discuss individual traces of work remaining and their deviation from the means, we also calculated a statistic based on 100 separate and independent simulation runs and visualized them in the form of an error list plot. This plot is given in **Figure 2**. It shows the mean values of work remaining as note points for each time step, and the 95% confidence intervals as error bars.

Figure 2 clearly shows the "sawtooth" behavior of finished work that is put in hold state by the team responsible for module design and integration testing and released to the first team processing component-level task 1 at weeks 4, 8 etc. Interestingly, this periodically correlated effect does not spill over to the second component-level task and the module-level task under the given boundary conditions. Because of the higher task processing rates the processing of these tasks is relatively smooth. In **Figure 2** it can also be seen that the predictability of the work remaining continually decreases over time as the 95% confidence intervals become wider and wider.

Additional results of the simulation study concerning the identifiability of the reference model and the associated parameter uncertainty are shown in **Figure 3**. In this figure the root mean square deviation between the reference model matrices $\Phi_1(s)$ and $\Phi_1(1)$ and their least-squares estimates are shown. The root mean square deviations between the matrices show a very similar pattern: The more data points T are available, the smaller the root mean square deviation and the longer the release period s of finished work that was put in a hold state for a given time series length, the smaller the deviation. Furthermore, the 95% confidence intervals show that the more data points are available for estimation and the longer the release period, the smaller the confidence interval and therefore the more certain we can be about the estimates.

5. Conclusions

In conclusion, one can say that in the light of the standards of organizational modeling and simulation, the identifiability of the reference model of cooperative work in a simulated vehicle door development project is high and the uncertainty in parameter estimation is low. The results are highly consistent and replicable. The estimation technique is numerically efficient and shows a very good stability. We also investigated the identifiability of more complex reference models and the resulting parameter uncertainty in additional simulation runs based on the same experimental design and the same methods. The additional results show that even the higher-dimensional reference models can be accurately identified from data, if constraints are posed on the auxiliary variable ε. However, if the auxiliary variable is also considered as a free parameter in least-squares estimation, the model´s identifiability is not good. Therefore, we recommend encoding all constraints of the developed PVAR model formulation in the description of the regression coefficient matrix R and the intercept vector b from linear relation *(9)* that are known in advance. If this is not possible, we recommended reducing the number of independent parameters by setting selected entries in the WTMs $\Phi_1(s)$ and $\Phi_1(1)$ to zero as Ursu and Duchesne *(2009)* have successfully shown for economic time series. Otherwise, one must expect very large sample sizes to be required to accurately estimate the true representation from data.

authors


Christopher M. Schlick was born in Germany on July 1, 1967. He received the M.S. degree (Dipl.-Ing.) in Electrical Engineering from Berlin University of Technology in 1992, Ph.D. degree (Dr.-Ing.) in Mechanical Engineering from Aachen University of Technology in 1999, and the Habilitation degree (Dr.-Ing. habil) also in Mechanical Engineering from Aachen University of Technology in 2004. He worked for the computer industry in 1992 and 1993 as a design engineer. From 1994 to 2000, he joined the Institute of Industrial Engineering and Ergonomics at RWTH Aachen University of Technology. From 2000 to 2004 he was the head of department of human–machine systems at the Research Institute for Communication, Information Processing and Ergonomics, Wachtberg, Germany. He is now a full professor of industrial engineering and ergonomics at RWTH Aachen University and deputy director of the Fraunhofer Institute for Communication, Information Processing and Ergonomics FKIE. His current interests are mathematical models of cooperative work in development projects and information-theoretic methods to evaluate emergent complexity.


Sebastian Terstegen received the M.Sc. (Dipl.-Ing.) degree in Electrical Engineering from the University of Paderborn, Germany, in 2009. He currently works as a PhD student and research assistant at the Institute of Industrial Engineering and Ergonomics of RWTH Aachen University. His research focuses on the development of simulation and optimization techniques for project management.


Sönke Duckwitz received the Dipl.-Ing. and the Dipl.-Wirt.Ing. degrees from RWTH Aachen University, Germany. He is currently head of department work organization and chief engineer at the Institute of Industrial Engineering and Ergonomics at RWTH Aachen University. His research interests include project management and modeling and simulation of complex working processes.

references

Eppinger, S.D., Browning, T. *(2012).* Design Structure Matrix Methods and Applications. MIT Press, Cambridge, MA.

Huberman, B. A., Wilkinson, D. M. *(2005).* Performance Variability and Project Dynamics. Computational and Mathematical Organization Theory 11 *(4)*, 307–332.

Lütkepohl, H. *(2005).* New Introduction to Multiple Time Series Analysis. Springer, Berlin.

Mihm, J., Loch, C. *(2006).* Spiraling Out of Control: Problem-Solving Dynamics in Complex Distributed Engineering Projects. In: Braha, D.; Minai, A.A.; Bar-Yam, Y. *(eds.)*. Complex Engineered Systems: Science Meets Technology. Springer, Berlin, 141–158.

Schlick, C.M., Schneider, S., Duckwitz, S. *(2011).* Modeling of Periodically Correlated Work Processes in Large-scale Concurrent Engineering Projects based on the DSM. Proceedings of the 13th International Dependency and Structure Modeling Conference, DSM 2011, 273–290.

Schlick, C., Duckwitz, S., Schneider, S. *(2013).* Project Dynamics and Emergent Complexity. Comput Math Organ Theory 19 *(1)*, 480–515.

Smith, R.P., Eppinger, S.D. *(1997).* Identifying Controlling Features of Engineering Design Iteration. Management Science 43 *(3)*, 276–293.

Ursu, E., Duchesne, P. *(2009).* On Modelling and Diagnostic Checking of Vector Periodic Autoregressive Time Series Models. Journal of Time Series Analysis 30 *(1)*, 70–96.

Yassine, A. A., Joglekar, N., Braha, D., Eppinger, S. D., & Whitney, D. *(2003).* Information Hiding in Product Development