13TH INTERNATIONAL DEPENDENCY AND STRUCTURE MODELLING CONFERENCE, DSM'11
CAMBRIDGE, MASSACHUSETTS, USA, SEPTEMBER 14 – 15, 2011

MODELING OF PERIODICALLY CORRELATED WORK PROCESSES IN LARGE-SCALE CONCURRENT ENGINEERING PROJECTS BASED ON THE DSM

Christopher M. Schlick, Sebastian Schneider and Sönke Duckwitz
Institute of Industrial Engineering and Ergonomics, RWTH Aachen University

ABSTRACT
This paper presents theoretical analyses of project dynamics and emergent complexity in large-scale new product development projects that are subject to the management concept of concurrent engineering. A model-driven approach is taken and a mathematical model of cooperative task processing is formulated on the basis of the theory of stochastic periodic vector-autoregressive processes. The model can capture not only the dynamic processing of the development tasks with short iteration length but also the long-scale effects of withholding the release of information on purpose. The model also provides the basis for the calculation of a closed-form solution for a metric of emergent complexity. The metric was invented in basic research and allows an explicit complexity assessment based on the model's independent parameters.

Keywords: project dynamics, cooperative work, complexity management, work transformation matrix

1 INTRODUCTION

In times of economic and financial revival after a great crisis, successful development of innovative products and effective management of new product development (NPD) projects are important for gaining competitive advantage. To shorten time-to-market, NPD projects are often subjected to concurrent engineering (CE). Winner et al. (1988) define CE as "a systematic approach to the integrated, concurrent design of products and their related processes, including manufacture and support. This approach is intended to cause the developers, from the outset, to consider all elements of the product life cycle from conception through disposal, including quality, cost, schedule, and user requirements." A good example is a vehicle development project in the automotive industry. In the late development stage, such a large-scale project involves hundreds of engineers collaborating in dozens of CE teams. The CE teams are structured according to the modules and components of the product to be developed and are coordinated by system-integration and management teams. The needs and requirements are "orchestrated" by the subject-matter experts in team meetings and mapped onto design parameters in a highly cooperative process on different time scales. Hence, these projects are truly "creative" and can show complex patterns of organizational dynamics. The patterns emerge from the periodically correlated work processes with continuing analysis, synthesis and decision-making stages and tight informational coupling through the product structure. As a consequence, the tasks are both highly variable and strongly dependent on each other and on elements of "surprise" in the form of seemingly erratic evolutionary events that occur. Variability and dependability render the project difficult to control, because a large body of knowledge of prior history is necessary to make good predictions, and the evolution toward a stable design solution can differ significantly from the expected (unperturbed) process (Huberman and Wilkinson, 2005). Depending on the intensity of the cooperative relationships in question, the development teams can enter vicious cycles of repeated revisions, which demand a lot of unplanned effort (Huberman and Wilkinson, 2005). The cycles can be reinforced, and a fatal pattern termed "design churns" (Yassine et al., 2003) can emerge. In this case, the project irregularly oscillates between being on, ahead of, or behind schedule. Design churns are an intriguing example of emergent complexity in NPD projects, which can lead to disastrous results. A deeper understanding of the interrelationships between performance variability and project dynamics is needed to cope with emergent complexity, together with new methods for quantitative analysis and evaluation.

2 DYNAMICS OF LARGE-SCALE CE PROJECTS

To analyze the interrelationships between project dynamics and emergent complexity, a model of cooperative task processing in large-scale CE projects is formulated. According to our previous work (Schlick et al., 2008, Schlick, 2011) the dynamics of a project with *p* fully concurrent but interacting tasks can be represented by a vector autoregression (VAR) model as

$$X_t = A_0 \cdot X_{t-1} + \varepsilon_t \qquad t \geq 1. \tag{1}$$

In the model, the random variable $X_t \in [0;1]^p$ represents the work remaining for all *p* tasks at time step *t*. The amount of work remaining can be measured by the time left to finalize a specific design or the number of open issues that need to be addressed before design release (Yassine et al., 2003). $A_0 = (a_{ij})$ is the $p \times p$ Work Transformation Matrix (WTM). The WTM is a task-oriented variant of the well known design structure matrix. Given a project phase, it is assumed that the WTM does not vary with time. The diagonal elements a_{ii} account for different productivity levels. They indicate the part of the work left incomplete after a (short) iteration over task *i* and therefore are defined on the interval]0;1[. The off-diagonal elements $a_{ij}\ (i \neq j)$ indicate the intensity and nature of cooperative relationships. Depending on their value, they have different meanings: 1) if $a_{ij} = 0$, work carried out on task *j* has no direct effect on task *i*; 2) if $a_{ij} > 0$, work on task *j* slows down the processing of task *i*, and one unit of work on task *j* at time step *t* generates a_{ij} units of extra work on task *i* at time step *t* + 1; 3) if $a_{ij} < 0$, work on task *j* accelerates the processing of task *i*, and one unit of work on task *j* reduces the work on task *i* by a_{ij} units at time step *t* + 1. This paper only considers phases of large-scale CE projects in which subgroups of interacting tasks must be processed in parallel. This means that no task in the subgroup is processed independently of the others, because input by other tasks is required regularly. *t* = 0 indicates the beginning of a project phase. It is often assumed that all parallel 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 random variable ε_t is added to model performance fluctuations. In CE projects there are many performance-shaping factors. Although we do not know their exact number or distribution, the central limit theorem tells us that, to a large degree, the sum of independently and identically distributed factors can be represented by a Gaussian distribution $N(x;\mu,C)$ with location $\mu = E[x]$ and covariance $C = E[(\mu - x)(\mu - x)^{\mathrm{T}}]$. We assume that the fluctuations have no systematic component and that $\mu = (0 \quad 0 \quad \ldots \quad 0)^{\mathrm{T}}$. Hence, the noise term in the state eq. (1) is expressed by $\varepsilon_t = N(x;0,C)$. The formulated model of cooperative task processing is closely related to the dynamical model of product development on complex directed networks which was introduced by Braha and Bar-Yam (2007). However, there are some subtle differences: 1) the VAR model is defined over a continuous range of state values and can therefore represent different kinds of cooperation relationships as well as precedence relations (e.g. overlapping); 2) each task is non-equally influenced by other tasks; 3) organization-induced correlations ρ_{ij} between performance fluctuations among tasks *i* and *j* can be captured. These correlations often have a strong effect on the course of the project. To reinforce the correlations, the covariance matrix *C* must have nonzero off-diagonal elements (nonisotropic noise).

A logical extension of the developed VAR model is to formulate a so-called periodic vector autoregressive (PVAR) stochastic process (Ursu and Duchesne, 2008). A PVAR model can capture not only the processing of the development tasks with short iteration length but also long-scale effects of withholding the release of information on purpose. Short iterations for a given amount of work are necessary to process component-level design information within development teams as already shown above. Frequent iterations between teams carrying out component level and system-level design are also necessary if the scope of predictability for the development project is small and only a few stable assumptions can be made about design ranges or physical functions. In fact, the organization of the problem-solving processes is such that, by definition, the tasks are cooperatively processed, in the sense that information about their individual progress cannot be hidden. An additional long-scale effect, however, often occurs in large-scale CE projects with hierarchical structures (Yassine et al., 2003), because system-level teams may withhold the release of a certain fraction of information about integration and tests of geometric/topological entities for a limited period. Between the releases, new information is “hidden” (kept secret), and work in the subordinate teams is based on the old state of knowledge. Such a hold-and-release policy is typical for CE projects in the automotive industry. This kind of “noncooperative behavior” is justified by the desire to improve solution maturity and reduce coordination. A deterministic model capable of capturing both cooperative and noncooperative task processing was developed by Yassine et al. (2003). In their paper, a time-variant state equation was formulated and validated based on simulation runs. We built directly upon their results in the following. However, the PVAR approach can also account for unpredictable fluctuations in performance and can be the basis for analytical complexity evaluations (see eq. 9). To formulate the PVAR model, it is assumed that a certain amount of finished system-related work is released by the development teams responsible for system integration and testing to component-level teams only, at time step ns ($n \in \mathbf{N}, s \geq 2$). At all other time steps, $ns+v$ ($n = 0,1,...; v = 1,...,s-1$) the tasks are processed by short iterations without withholding information. Under these assumptions, the state eq. 1 can be generalized to a process Y_t with periodically correlated components:

$$Y_{ns+v} = \Phi_1(v) \cdot Y_{ns+v-1} + \varepsilon_{ns+v} \qquad (2)$$

where the index n indicates the long time scale with period s, and v the short time scale. $Y_t = (Y_t(1),...,Y_t(d))^{\mathrm{T}}$ is a $d \times 1$ single random vector encoding the state of the project at time step $t = ns+v$. The leading components of the state vector represent the work remaining of the $p^{\mathrm{C}} \in \mathbf{N}$ component-level and $p^{\mathrm{S}} \in \mathbf{N}$ system-level tasks that are processed on the short time scale. For these tasks the work transformation can be captured by a combined WTM A_0 as

$$A_0 = \begin{pmatrix} A_0^C & A_0^{SC} \\ A_0^{CS} & A_0^S \end{pmatrix}. \qquad (3)$$

In the combined WTM the submatrix A_0^C of size $p^{\mathrm{C}} \times p^{\mathrm{C}}$ is the dynamical operator for the cooperative processing of component-level tasks. The $p^{\mathrm{S}} \times p^{\mathrm{S}}$ submatrix A_0^S refers to system-level tasks in an analogous manner. The $p^{\mathrm{C}} \times p^{\mathrm{S}}$ submatrix A_0^{SC} determines the rework fraction created by system-level tasks for the corresponding component-level tasks, whereas the $p^{\mathrm{S}} \times p^{\mathrm{C}}$ submatrix A_0^{CS} determines the rework fraction created by component-level tasks for the system-level tasks. Moreover, the sub-states $(Y_t(1),...,Y_t(p^{\mathrm{C}}+p^{\mathrm{S}}))$ have to be augmented by other p^{S} sub-states to account for the periodic hold-and-release policy. The augmented p^{S} sub-states do not represent the work remaining as the leading states do, but represent 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. Through the model formulation the finished work can be placed in hold state. The associated $p^{\mathrm{S}} \times p^{\mathrm{S}}$ submatrix A_0^{SH} captures the rework fraction created by the system-level tasks in each iteration at time step $v = 1,...,s-1$. After release additional rework is generated for the component-level tasks. This rework is calculated based on the WTM A_0^{HC}. This WTM is of size $p^{\mathrm{C}} \times p^{\mathrm{S}}$. There, $d = p^{\mathrm{C}} + 2p^{\mathrm{S}}$ holds. The periodically correlated work processes are represented by the time evolution of the state vector Y_{ns+v} based on the coefficients $\Phi_1(v)$. The two time scales correspond to indices n and v. The long time scale is indexed by n. In seasonal macroeconomic models, for instance, n indicates the year that the time series refer to (e.g. consumption). Clearly, in large-scale CE projects the release period is much shorter and typically covers intervals of four to eight weeks. The short time scale is indexed by v. On this scale the iterations usually occur on a daily or weekly basis. In the terminology of macroeconomic models v indicates a “season” of the “year.” The length s of the period between release of hidden information has to be predetermined. For a period length s, the random vector Y_{ns+v} contains the realization of work remaining during the vth iteration over all component-level and system-level tasks at the release period n and the amount of finished work on system level that is ready to be released to component-level tasks in period $n+1$. Furthermore, the periodically correlated task processing has to be modeled, and for this aim two dynamical operators are introduced (Yassine et al., 2003). These operators determine the autoregressive coefficients $\Phi_1(v)$ in state eq. 2. The release of hidden information over s time steps is modeled by the first dynamical operator $\Phi_1(s)$. It is assumed that the release occurs at the end of the period.

The operator $\Phi_1(s)$ can be composed as

$$\Phi_1(s)=\begin{pmatrix} A_0^C & A_0^{SC} & A_0^{HC} \\ A_0^{CS} & A_0^S & 0 \\ 0 & 0 & \{\varepsilon\}\cdot I_{p^S} \end{pmatrix}. \tag{4}$$

In the above equation the ε-symbol denotes an arbitrarily small positive quantity. The definition of nonzero interactions between the augmented p^S sub-states is necessary to explicitly evaluate the emergent complexity of the modeled CE project. We will return to this point in the next section. For practical purposes, it is recommended to calculate with $\varepsilon = 0.001$. By doing so, the finished work after release is set back to a nonzero but negligible amount. The task processing in the $v = 1,\ldots,s-1$ iterations before release is modeled on the basis of a second dynamical operator $\Phi_1(1)$. In contrast to macroeconomic models, it is assumed that the autoregressive coefficients are constant during one period s, that is $\Phi_1(1) = \ldots = \Phi_1(s-1)$. $\Phi_1(1)$ can be composed of the previously defined submatrices in an analogous manner as

$$\Phi_1(1)=\begin{pmatrix} A_0^C & A_0^{SC} & 0 \\ A_0^{CS} & A_0^S & 0 \\ 0 & A_0^{SH} & \{1-\varepsilon\}\cdot I_{p^S} \end{pmatrix}. \tag{5}$$

Finally, the combined error process ε_{ns+v} corresponds to a zero-mean periodic white noise. That means ε_{ns+v} is composed of $d\times 1$ random vectors representing unpredictable performance fluctuations such that $E[\varepsilon_t]=0$ and $E[\varepsilon_{ns+v}\,\varepsilon_{ns+v}{}^{\mathrm{T}}]=C$. It is assumed that the covariance matrix C is not singular. An important result from the theory of stochastic processes (Ursu and Duchesne, 2008) is that a PVAR process offers a compact representation as a VAR model:

$$\Phi_0^*\cdot Y_n^* = \Phi_1^*\cdot Y_{n-1} + \varepsilon_n^* \tag{6}$$

where $Y_n^* = \left(Y_{ns+s}^{\mathrm{T}}, Y_{ns+s-1}^{\mathrm{T}},\ldots,Y_{ns+1}^{\mathrm{T}}\right)^{\mathrm{T}}$ and $\varepsilon_n^* = \left(\varepsilon_{ns+s}^{\mathrm{T}}, \varepsilon_{ns+s-1}^{\mathrm{T}},\ldots,\varepsilon_{ns+1}^{\mathrm{T}}\right)^{\mathrm{T}}$ are $ds\times 1$ state and error vectors. The nonsingular matrix Φ_0^* and the autoregressive coefficient Φ_1^* are given by

$$\Phi_0^*=\begin{pmatrix} I_d & -\Phi_1(s) & 0 & \cdots & 0 & 0 \\ 0 & I_d & -\Phi_1(s-1) & \cdots & 0 & 0 \\ \vdots & & & \ddots & & \vdots \\ 0 & 0 & 0 & \cdots & I_d & -\Phi_1(1) \\ 0 & 0 & 0 & \cdots & 0 & I_d \end{pmatrix} \tag{7}$$

and

$$\Phi_1^*=\begin{pmatrix} \Phi_s(s) & \Phi_{s+1}(s) & \cdots & \Phi_{s+s-1}(s) \\ \Phi_{s-1}(s-1) & \Phi_s(s-1) & \cdots & \Phi_{s+s-2}(s-1) \\ \vdots & & \ddots & \vdots \\ \Phi_1(1) & \Phi_2(1) & \cdots & \Phi_s(1) \end{pmatrix}, \tag{8}$$

where $\Phi_k(v)=0$ for $k > 1$ and $\Phi_1(v)=\Phi_1(1)$ for $v = 1\ldots s-1$. The matrices Φ_0^* and Φ_1^* are both of size $ds\times ds$. It can be proven that the PVAR stochastic process is convergent and the work in the modeled project does not grow over all given limits if $\mathrm{Det}\left(I_d-\Phi_1(s)\left(\Phi_1(1)\right)^{s-1}z\right)\neq 0$ for all $z\in\mathbf{C}$ such that $|z|<1$. If the process is causal, stationarity results for VAR models can be invoked easily (Lütkepohl, 2005). To evaluate emergent complexity explicitly (section 3), the process must satisfy the criterion of strict stationarity. A strictly stationary process has a joint probability density that is invariant under shifting the origin. To guarantee strict stationarity, examination of the eigenvalues of $\Phi_1(s)\left(\Phi_1(1)\right)^{s-1}$ to check that the magnitudes are all strictly smaller than one is sufficient. The parameters of a PVAR process can be calculated efficiently on the basis of least-square or maximum-likelihood estimation (see Ursu and Duchesne, 2008). Building a parameterized model fulfils manifold purposes: 1) to get a deeper understanding of cooperative work in NPD; 2) to capture the essential characteristics of project dynamics influencing productivity, stability, emergent complexity and other key performance indicators; 3) to be able to formulate a small set of concepts that can be used for organizational design and optimization and 4) to make good predictions about the “evolution” of the project over time. In other words, the model helps to manage complex work processes and to make substantiated decisions based on well-understood and explicitly formulated essentials of the modeled CE project.

3 EVALUATION OF EMERGENT COMPLEXITY

Concerning the second of the abovementioned purposes, we developed a metric for the evaluation of emergent complexity in CE projects in previous work. The metric is based on the complexity theory of the theoretical physicist Peter Grassberger (1986). Following Grassberger’s terminology we call the metric “Effective Measure Complexity” (EMC). The interested reader can find details in Schlick et al. (2008) and Schlick (2011). Generally speaking, the metric is a lower bound of the amount of information required for optimal prediction of the state of a complex project evolving over time. EMC can be calculated either on the basis of an explicit project model, as we do in this paper, or from project data alone, without intervening models. Since it can quantify the degree of “informational structure” between the past and the future, it is especially interesting for the evaluation of CE projects. The derivation of the metric based on state eqs. (2) and (6) in conjunction with the definitions according to eqs. (4), (5), (7) and (8) is mathematically very involved and not presented in this paper. We only give the closed-form solution as a function of the release period *s*:

$$EMC(s) = \frac{1}{2}\log_2\left(\frac{\mathrm{Det}\left(\sum_{k=0}^{\infty}\left(\Phi_0^{*-1}\cdot\Phi_1^*\right)^k\cdot\left(\Phi_0^{*-1}\cdot C^*\cdot\left(\Phi_0^{*-1}\right)^{\mathrm{T}}\right)\cdot\left(\Phi_1^{*\mathrm{T}}\cdot\left(\Phi_0^{*-1}\right)^{\mathrm{T}}\right)^k\right)}{\mathrm{Det}\left(\Phi_0^{*-1}\cdot C^*\cdot\left(\Phi_0^{*-1}\right)^{\mathrm{T}}\right)}\right) \tag{9}$$

In the above equation the matrix C^* is the covariance matrix of the composed random vector ε_n^*.

ACKNOWLEDGEMENTS

The research was funded by the Federal Ministry of Education and Research (BMBF) according to grant number 01IS10038B.

REFERENCES

Braha, D., & Bar-Yam, Y. (2007). The Statistical Mechanics of Complex Product Development: Empirical and Analytical Results. *Management Science*, *53*(7), 1127–1145.

Grassberger, P. (1986). Toward a Quantitative Theory of Self-Generated Complexity. *International Journal of Theoretical Physics*, *25*(9), 907–938.

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*. Berlin: Springer.

Schlick, C., Duckwitz, S., Gärtner, T., & Schmidt, T. (2008). A complexity measure for concurrent engineering projects based on the DSM. In: *Proceedings of the 10th International DSM Conference*, Stockholm, November 2008, 219–230.

Schlick, C. (2011). *Project dynamics and Emergent Complexity*. RWTH Aachen University, Institute of IEE, Working Paper 11-01-1C. arXiv:1101.0754v2 [nlin.AO]. http://arxiv.org/abs/1101.0754.

Ursu, E., & Duchesne, P. (2008). On Modelling and Diagnostic Checking of Vector Periodic Autoregressive Time Series Models. *Journal of Time Series Analysis*, 30 (1), 70–96.

Winner, R. I., Pennell, J. P., Bertrand, H. E., & Slusarezuk, M. M. (1988). *The Role of Concurrent Engineering in Weapons System Acquisition*. IDA-Report R-338, Institute for Defense Analyses; Alexandria, VA.

Yassine, A., Joglekar, N., Braha, D., Eppinger, S. D., & Whitney, D. (2003). Information Hiding in Product Development: The Design Churn Effect. *Research in Engineering Design*, *14*(3), 145−161.

Contact: Christopher M. Schlick, RWTH Aachen University, Institute of Industrial Engineering and Ergonomics, Bergdriesch 27, 52062 Aachen, Germany, +49 241 80 99 440, c.schlick@iaw.rwth-aachen.de

INVEST ON VISUALIZATION

Modeling of Periodically Correlated Work Processes in Large-Scale Concurrent Engineering Projects Based on the DSM

C. M. Schlick, S. Schneider and S. Duckwitz
Institute of Industrial Engineering and Ergonomics
RWTH Aachen University, Germany

Technische Universität München

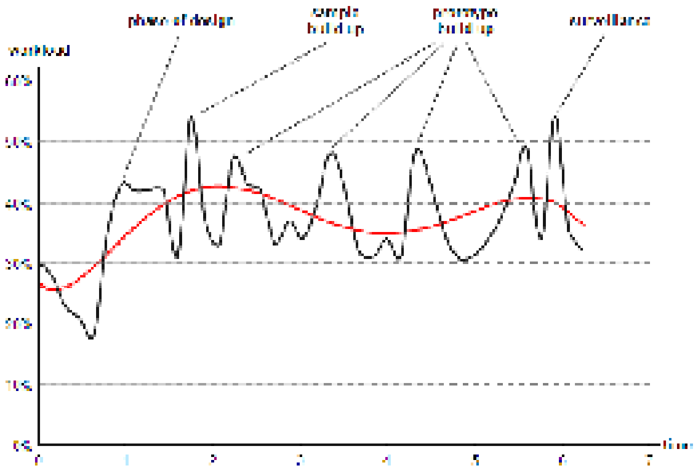
INVEST ON VISUALIZATION

Index

- Complexity in large-scale concurrent engineering projects
- Basic dynamic model of cooperative task processing
- Extended dynamic model with periodically correlated work processes
- Analysis of project dynamics – a simulation study
- Complexity assessment in open dynamical systems
- Closed-form solution of complexity metric as well as productivity metric for periodically correlated work processes
- Outlook

Complexity in Large-Scale Concurrent Engineering Projects

- Structural complexity: large number of engineers in multifunctional teams who make partially autonomous design decisions, but are tightly coupled through the product structure with many interfaces between mechanical, electronic and software components
- Behavioral / emergent complexity: Regular and irregular iterations on different time scales due to availability of new information about geometric/topological entities in conjunction with non-predictable performance fluctuations; excitation through feed-forward and feedback loops can lead to “design churn effects” (Yassine et al. 2003)



oscillating workload of design engineer in a development project of a multi-project environment in the German automotive industry

Dynamic Model of Cooperative Task Processing in Concurrent Engineering Projects

Vector Autoregression Model: $X_t = \mathbf{A}_0 \cdot X_{t-1} + \varepsilon_t \qquad X_t \in \Re^p \qquad \varepsilon_t = N(x;0,C)$

Schlick et al. (2008), Schlick (2011)

X_t : work remaining for all p tasks at time step t

$\mathbf{A}_0 = (a_{ij}) : p \times p$ Work Transformation Matrix (WTM)

ε_t : noise term representing unpredictable performance fluctuations

Design Structure Matrix

Work Transformation Matrix (WTM)

Eppinger & Smith (1997), Huberman & Wilkinson (2005)

task

A develop component 1

B develop component 2

C develop component 3

D integration testing

E ...

$$\begin{array}{ccccc} A & B & C & D & E \\ (X) & & & X & \dots \\ X & (X) & & X & \\ X & X & (X) & X & \\ X & X & X & (X) & \\ \dots & & & & \end{array} \longrightarrow \mathbf{A_0} = \begin{pmatrix} 0.95 & 0 & 0 & 0.01 \\ 0.20 & 0.90 & 0 & 0.02 \\ 0.08 & 0.04 & 0.87 & 0.05 \\ 0.05 & 0.02 & 0.02 & 0.93 \end{pmatrix}$$

a_{ii} : autonomous task processing rates

a_{ij} : “coupling strength” of tasks, generating rework

Extended Dynamic Model of Task Processing in Large-Scale Concurrent Engineering Projects (I)

Assumptions

- Tasks are processed on short and long time scales. Short iterations are necessary to quickly process component-level and system-level design information within teams. Long iterations occur because system-level teams may withhold release of information for a limited period.
- Tasks are processed cooperatively (without withholding information) on short time scale at time step $ns + v$ $(n = 0,1,\dots;\, v = 1,\dots,s-1)$.
- Task are processed “noncooperatively” on long time scale, i.e. certain amount of finished work about integration testing is released by system-level teams only at time step ns $(n \in \mathbf{N}, s \geq 2)$.

Vector Autoregression Model: $X_t = \mathbf{A}_0 \cdot X_{t-1} + \varepsilon_t$

Schlick et al. (2008), Schlick (2011)

Periodic Vector Autoregression Model: $Y_{ns+v} = \mathbf{\Phi}_1(v) \cdot Y_{ns+v-1} + \varepsilon_{ns+v}$ $\quad Y_{ns+v} \in \Re^d \quad \varepsilon_{ns+v} = N(x;0,C)$

n: indicates long time scale with period s

v: indicates short time scale

$Y_t = (Y_t(1),\dots,Y_t(d))^T$: $d \times 1$ random vector encoding project state at time step $t = ns + v$

$\mathbf{\Phi}_1(v)$: autoregressive coefficients

ε_{ns+v}: extended noise term

C: $d \times d$ covariance matrix of noise

INVEST ON VISUALIZATION

Extended Dynamic Model of Task Processing in Large-Scale Concurrent Engineering Projects (II)

Model leads to periodically correlated task processing based on autoregressive coefficients

$$\mathbf{\Phi}_1(v) = \mathbf{\Phi}_1(1) = \begin{pmatrix} \mathbf{A}_0^{C} & \mathbf{A}_0^{SC} & 0 \\ \mathbf{A}_0^{CS} & \mathbf{A}_0^{S} & 0 \\ 0 & \mathbf{A}_0^{SH} & \{1-\varepsilon\} \cdot I_{p^s} \end{pmatrix}$$

task processing in the $v = 1,\dots,s-1$ iterations before release

$$\mathbf{\Phi}_1(s) = \begin{pmatrix} \mathbf{A}_0^{C} & \mathbf{A}_0^{SC} & \mathbf{A}_0^{HC} \\ \mathbf{A}_0^{CS} & \mathbf{A}_0^{S} & 0 \\ 0 & 0 & \{\varepsilon\} \cdot I_{p^s} \end{pmatrix}$$

release of information that is put in hold state over s time steps

using a combined Work Transformation Matrix as dynamical operator

$$\mathbf{A}_0 = \begin{pmatrix} \mathbf{A}_0^{C} & \mathbf{A}_0^{SC} \\ \mathbf{A}_0^{CS} & \mathbf{A}_0^{S} \end{pmatrix}$$

combines four individual WTMs:

1) cooperative processing of component-level tasks $\mathbf{A}_0^{C}$
2) cooperative processing of system-level tasks $\mathbf{A}_0^{S}$
3) rework fraction created by system-level tasks for component-level tasks $\mathbf{A}_0^{SC}$
4) rework fraction created by component-level tasks for system-level tasks $\mathbf{A}_0^{CS}$.

e.g. $\mathbf{A}_0^{SC}$ $\Rightarrow$ system-level tasks; $\Downarrow$ component-level tasks

$$\begin{array}{c} \\ \text{C1} \\ \text{C2} \end{array} \begin{array}{c} \begin{array}{ccc} \text{S1} & \text{S2} & \text{S3} \end{array} \\ \begin{pmatrix} 0.06 & 0 & 0.10 \\ 0.20 & 0.10 & 0 \end{pmatrix} \end{array}$$

e.g. system-level task S1 creates 6% rework for component-level task C1 and 20% rework for tasks C2 at each short iteration

Extended Dynamic Model of Task Processing in Large-Scale Concurrent Engineering Projects (III)

plus two hold-and-release matrices related to integration testing

e.g. $\mathbf{A}_0^{SH}$ ⇒ system-level tasks

⇓ finished work placed in hold state

$$\begin{array}{cc} & \begin{array}{ccc} S1 & S2 & S3 \end{array} \\ \begin{array}{c} H1 \\ H2 \\ H3 \end{array} & \begin{pmatrix} 0.05 & 0 & 0 \\ 0 & 0.06 & 0 \\ 0 & 0 & 0.07 \end{pmatrix} \end{array}$$

e.g. system-level task S1 generates 5% of finished work at each short iteration that is put in hold state and is accumulated over all short iterations; the finished work remains hidden for the component-level teams until it is released at time step *ns* ; in an analogous manner system-level task S2 generates 6% and task S3 7% of finished work at each short iteration

$\mathbf{A}_0^{HC}$ ⇒ system-level tasks

⇓ component-level tasks

$$\begin{array}{cc} & \begin{array}{ccc} S1 & S2 & S3 \end{array} \\ \begin{array}{c} C1 \\ C2 \end{array} & \begin{pmatrix} 1.00 & 0 & 0.10 \\ 0 & 0.95 & 0 \end{pmatrix} \end{array}$$

e.g. after information release at time step *ns* the system-level task S1 generates additional work for component-level task C1 which is based on finished work that has been accumulated over the short iterations; this additional work amounts for 100% of the finished work; in an analogous manner system-level task S2 generates additional work for component-level task 2; in this case only 95% of finished work are relevant

Simulation of Project Dynamics (I)

In the following the task processing in an example project defined by a parametric state equation is simulated. The example project includes two component-level tasks and two system-level tasks. These tasks are processed by different teams.

Subject of the component level tasks is the design of an instrument panel and a center console of a vehicle (cf. McDaniel 1996). The system-level tasks deal with integration testing of these and other components. Both system-level tasks generate 3% of finished work at each short iteration that is put in hold state until it is released at time step *ns*.

The release period *s* is considered as an independent parameter. The parameter set is as follows:

$$\mathbf{A}_0^C = \begin{pmatrix} 0.90 & 0.05 \\ 0.05 & 0.90 \end{pmatrix} \quad \mathbf{A}_0^{CS} = \begin{pmatrix} 0.06 & 0 \\ 0 & 0 \end{pmatrix} \quad \mathbf{A}_0^{SH} = \begin{pmatrix} 0.03 & 0 \\ 0 & 0.03 \end{pmatrix}$$

$$\mathbf{A}_0^S = \begin{pmatrix} 0.85 & 0.07 \\ 0.07 & 0.85 \end{pmatrix} \quad \mathbf{A}_0^{SC} = \begin{pmatrix} 0.06 & 0 \\ 0 & 0 \end{pmatrix} \quad \mathbf{A}_0^{HC} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

$$\mathbf{\Phi}_1(v) = \begin{pmatrix} \mathbf{A}_0^C & \mathbf{A}_0^{SC} & 0 \\ \mathbf{A}_0^{CS} & \mathbf{A}_0^S & 0 \\ 0 & \mathbf{A}_0^{SH} & \{1 - \varepsilon\} \cdot I_2 \end{pmatrix} \quad \mathbf{\Phi}_1(s) = \begin{pmatrix} \mathbf{A}_0^C & \mathbf{A}_0^{SC} & \mathbf{A}_0^{HC} \\ \mathbf{A}_0^{CS} & \mathbf{A}_0^S & 0 \\ 0 & 0 & \{\varepsilon\} \cdot I_2 \end{pmatrix} \quad \varepsilon = 0.001$$

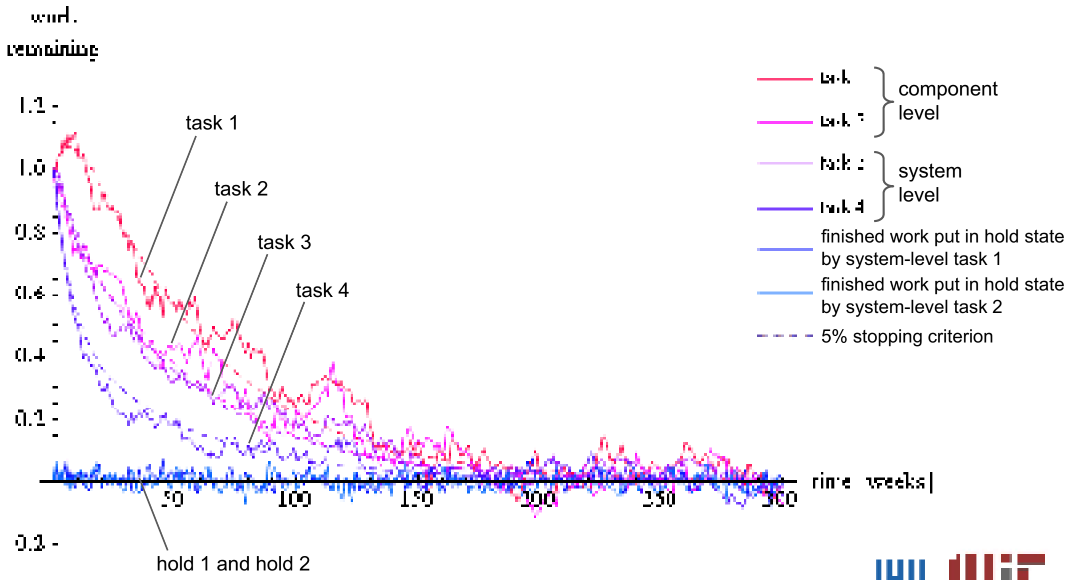
$$C = \{s\} \cdot \begin{pmatrix} \mathbf{\Phi}_1(1) \cdot I_2 \cdot \mathbf{\Phi}_1(1)^{\mathrm{T}} & 0 & \cdots \\ 0 & \ddots & 0 \\ \vdots & 0 & \mathbf{\Phi}_1(s) \cdot I_2 \cdot \mathbf{\Phi}_1(s)^{\mathrm{T}} \end{pmatrix}$$

$s = 0.02$: scalar value of variance of fluctuations

$I_2 : 2 \times 2$ identity matrix

Simulation of Project Dynamics (II)

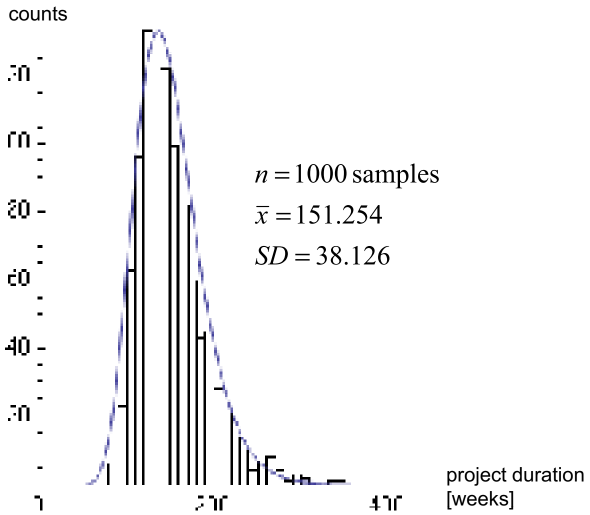
Shown below are simulated traces of work remaining for all development task that are processed on short and long time scales. The long time scale represents the hold-and-release policy. The release period is *s*=2.

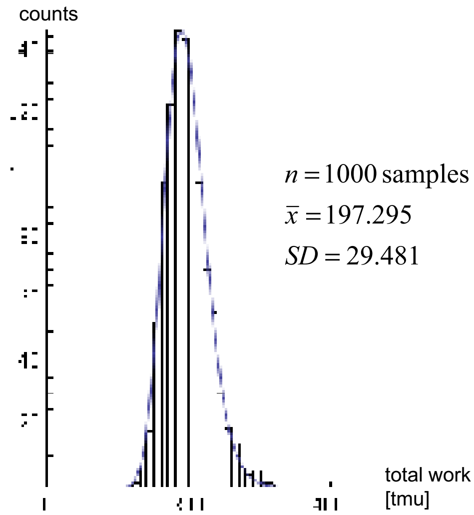


INVEST ON VISUALIZATION

Simulation of Project Dynamics (III)

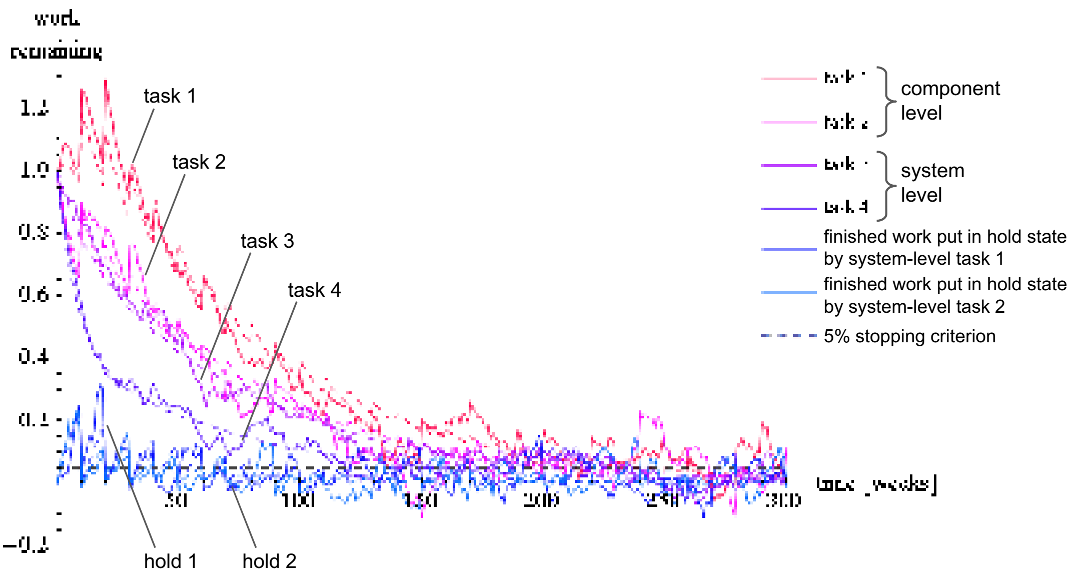
Corresponding histograms of project duration and total work for 1000 simulated CE projects (*s*=2). Stopping criterion was that at most 5% of work remained for all tasks. The total work is the total work remaining over all tasks and all time steps until the stopping criterion is met.





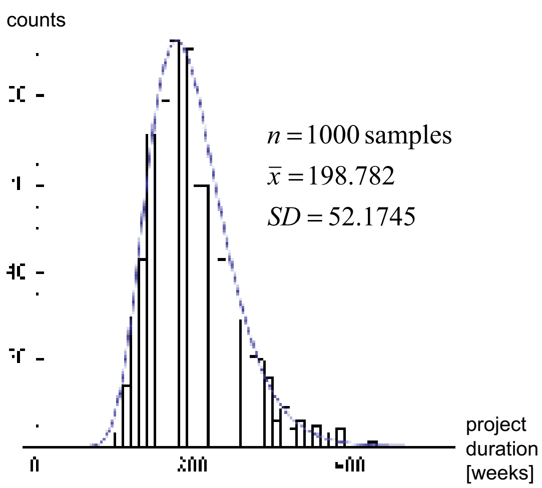
Simulation of Project Dynamics (IV)

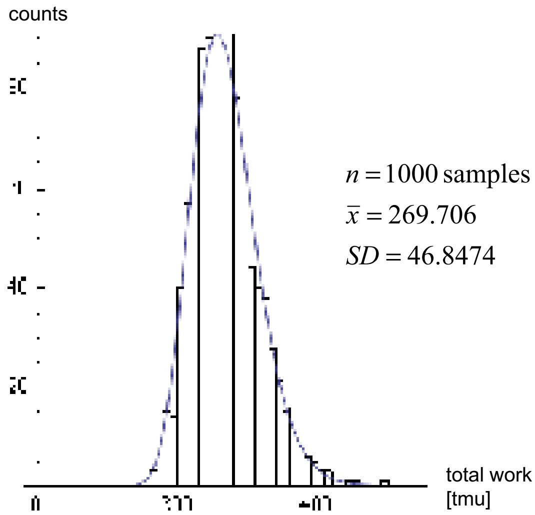
Shown below are additional traces of work remaining for an extended release period *s*=10.



Simulation of Project Dynamics (V)

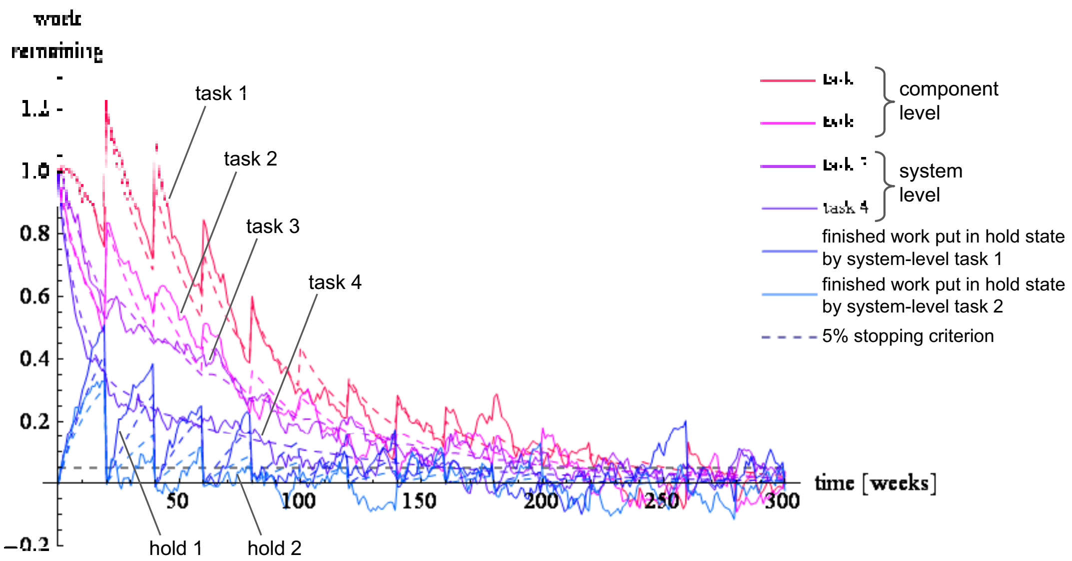
Corresponding histograms of project duration and total work for 1000 simulated CE projects (*s*=10). Stopping criterion was that at most 5% of work remained for all tasks.





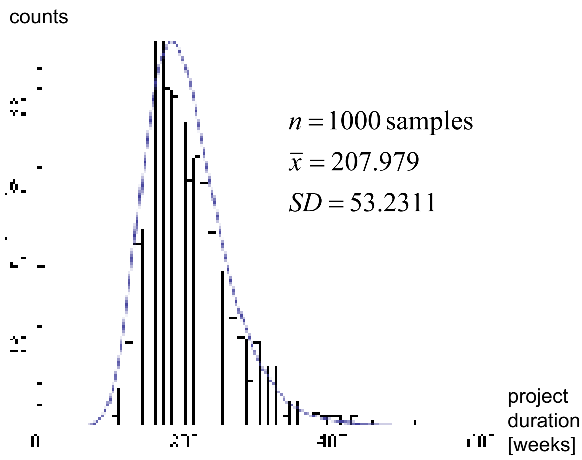
Simulation of Project Dynamics (VI)

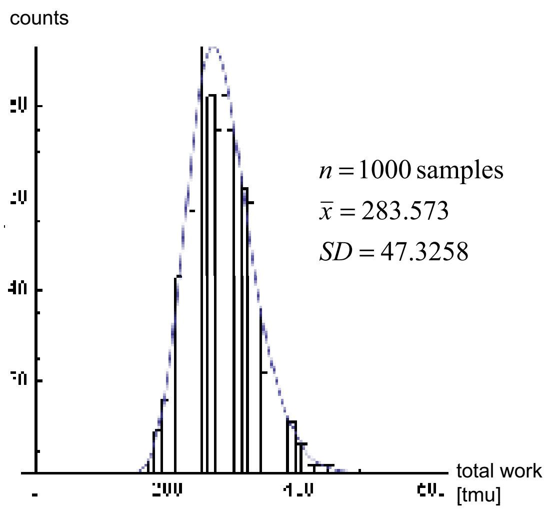
Shown below are additional traces of work remaining for an extended release period *s*=20.



Simulation of Project Dynamics (VII)

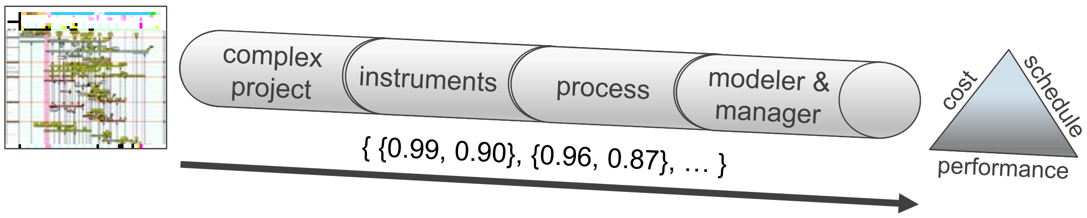
Corresponding histograms of project duration and total work for 1000 simulated CE projects (*s*=20). Stopping criterion was that at most 5% of work remained for all tasks.





Complexity Assessment in Open Dynamical Systems – An Information-Theoretic Metric

- P. Grassberger developed a seminal complexity theory for open dynamical systems
- His theory is the foundation for the calculation of DSM-based complexity metric for large-scale CE projects with periodically correlated work processes. Following Grassberger´s terminology the metric is called “Effective Measure Complexity” (EMC). EMC measures the mutual information between the past and future of a stochastic process. It is a lower bound of the unknown forecast complexity, which is the amount of information required to optimally predict system behavior. The approach can be visualized as a project pipeline:



- The central question to be answered by using the metric is: How much information that is generated by a complex project and measured through the instruments must be stored (in the presence), in order to predict the future course of the project as accurately as possible given information from the arbitrarily distant past?

Closed-Form Solution of Complexity Metric

The chosen model formulation allows to calculate the closed-form solution

$$EMC(s)=\frac{1}{2}\log_2\left(\frac{\mathrm{Det}\left(\sum_{k=0}^{\infty}\left(\mathbf{\Phi}_0^{*-1}\cdot\mathbf{\Phi}_1^{*}\right)^k\cdot\left(\mathbf{\Phi}_0^{*-1}\cdot C^{*}\cdot\left(\mathbf{\Phi}_0^{*-1}\right)\right)^T\cdot\left(\mathbf{\Phi}_1^{*T}\cdot\left(\mathbf{\Phi}_0^{*-1}\right)^T\right)^k\right)}{\mathrm{Det}\left(\mathbf{\Phi}_0^{*-1}\cdot C^{*}\cdot\left(\mathbf{\Phi}_0^{*-1}\right)^T\right)}\right)$$

with parameters

$$\mathbf{\Phi}_0^{*}=\begin{pmatrix} I_d & -\mathbf{\Phi}_1(s) & 0 & \cdots & 0 & 0 \\ 0 & I_d & -\mathbf{\Phi}_1(s-1) & \cdots & 0 & 0 \\ \vdots & & & \ddots & & \vdots \\ 0 & 0 & 0 & \cdots & I_d & -\mathbf{\Phi}_1(1) \\ 0 & 0 & 0 & \cdots & 0 & I_d \end{pmatrix}$$

C^{*} : covariance matrix of the composed random vector ε_n^{*}

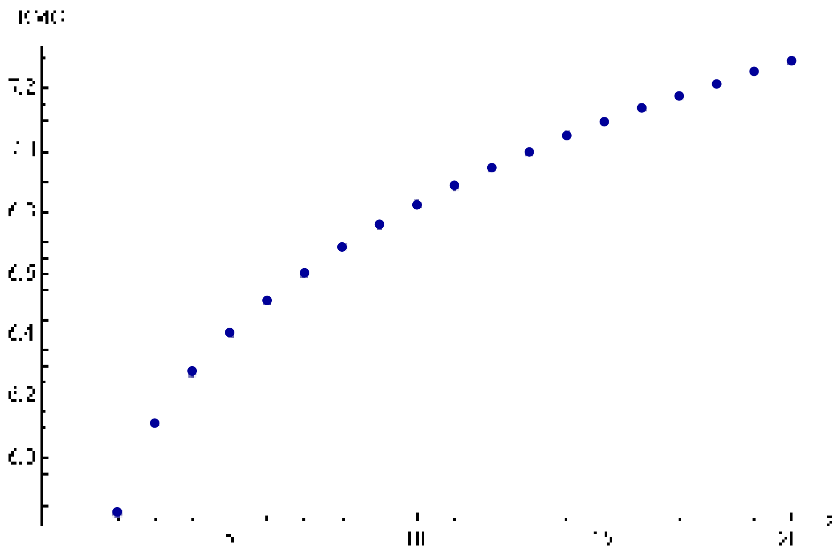
$$\mathbf{\Phi}_1^{*}=\begin{pmatrix} \mathbf{\Phi}_s(s) & \mathbf{\Phi}_{s+1}(s) & \cdots & \mathbf{\Phi}_{s+s-1}(s) \\ \mathbf{\Phi}_{s-1}(s-1) & \mathbf{\Phi}_s(s-1) & \cdots & \mathbf{\Phi}_{s+s-2}(s-1) \\ \vdots & & \ddots & \vdots \\ \mathbf{\Phi}_1(1) & \mathbf{\Phi}_2(1) & \cdots & \mathbf{\Phi}_s(1) \end{pmatrix}$$

where $\mathbf{\Phi}_k(v)=0$ for $k>1$ and

$\mathbf{\Phi}_1(v)=\mathbf{\Phi}_1(1)$ for $v=1,\ldots,s-1$

Values of Complexity Metric for Varying Period Length *s*

When the parameter set of the previous simulation study is used and the period *s* is varied systematically, the following complexity curve is obtained:

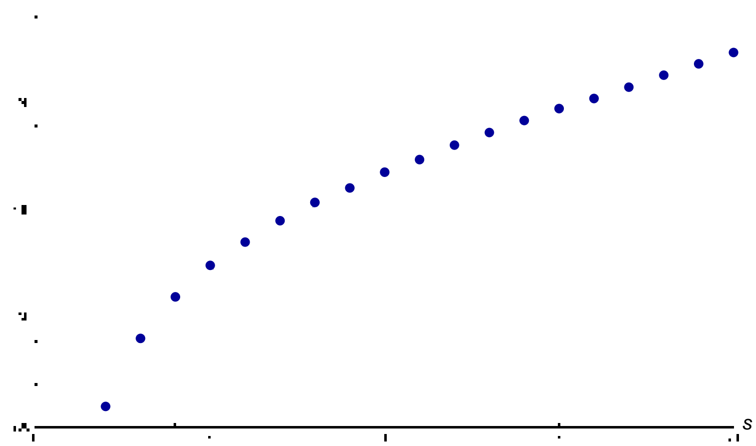


Closed-form Solution of Productivity Metric

Interestingly, the chosen model formulation also allows to calculate the expected total work that is generated in the CE project for $t \to \infty$ analytically. The closed-form solution of this productivity metric is:

$$TW(s) = \text{Total}\left(\left(I_d - \left(\boldsymbol{\Phi}_0^{*-1} \cdot \boldsymbol{\Phi}_1^{*}\right)\right)^{-1} \cdot x_0\right)$$

total work [tmu]



The function Total(.) calculates the sum of all components of the vector argument. The d 1 vector x_0 represents the initial work remaining. When the same parameters as in the previous simulation study are used and it is assumed that all development tasks are initially 100% to be completed, a total amount of work depending on *s* is generated as shown on the left hand side.

Outlook

- Future objective is to undertake external validation studies of the developed periodic vector autoregression model and the derived complexity metric with experienced project managers in industry.
- It is hypothesized that EMC is not only a conceptually valid complexity metric but also has the potential to capture the implicit knowledge of project managers based on the nature, quantity and magnitude of concurrent tasks and their interactions.
- An additional objective is to use EMC for analytical and simulation-based optimization of project organization design. Objective function is to minimize complexity subject to the constraint that the expected total work in the project is constant. Unfortunately, this constrained optimization problem is very hard to solve for projects with non-trivial cooperative relationships.
- Finally, it is planned to developed a generalized periodic vector autoregression model with latent variables. By doing so, it will be possible to distinguish between unpredictable performance fluctuations and uncertainty in project performance and progress evaluation.