Automatic Generation of Examinations in the Automatic Control Courses: Decision Support Matlab/LATEX Toolkit for Stepwise Constructive

Citation for the original published paper (version of record):
Stotsky, A., Wik, T. (2024). Automatic Generation of Examinations in the Automatic Control Courses: Decision Support
Matlab/LATEX Toolkit for Stepwise Constructive Alignment. IFAC-PapersOnLine, 58(16): 47-52.
http://dx.doi.org/10.1016/j.ifacol.2024.08.460

Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 58-16 (2024) 47–52

Automatic Generation of Examinations in the Automatic Control Courses: Decision Support Matlab/LaTeX Toolkit for Stepwise Constructive Alignment ⋆

Alexander Stotsky * and **Torsten Wik** **

* *Division of Systems & Control, Department of Electrical Engineering, Chalmers University of Technology, SE-412 96 Gothenburg, Sweden e-mail: alexander.stotsky@chalmers.se*
** *Division of Systems & Control, Department of Electrical Engineering, Chalmers University of Technology, SE-412 96 Gothenburg, Sweden e-mail: torsten.wik@chalmers.se*

Abstract: Final written examination is the most important part of summative assessment in automatic control courses. Preparation of the examinations with a given number of points according to the concept of Constructive Alignment (which could be the main concept in future automatic control education) takes significant amount of time of the educator and motivates development of a toolkit for automatic compilation of examination problems. A decision support Matlab/LaTeX toolkit based on random number generators for selection of examination problems is proposed in this paper to facilitate the alignment. The toolkit allows application of Stepwise Constructive Alignment (a new method proposed in this paper), where the alignment is achieved by a number of software runs associated with random trials. In each step the educator manually selects suitable problems before each run based on evaluation of the random choice from the previous run. Automatic generation of the examination together with solutions for the course 'Process control and measurement techniques' is presented as an example.

Keywords: Constructive Alignment in Automatic Control Education, Computer Aided Development of Educational Materials, Automatic Generation of Examinations with Desired Indices of Difficulty, Stepwise Constructive Alignment, Decision Support Matlab/LaTeX Toolkit

1. CONSTRUCTIVE ALIGNMENT IN AUTOMATIC CONTROL EDUCATION

Control systems are widely used in many application areas and therefore control education is an important part of most engineering programs. Control engineering is generally considered as a very challenging subject from many mathematical and practical aspects.
For improvement of the learning outcomes the concept of CA (Constructive alignment) developed by Biggs (1996) is widely applied in control courses, see for example Knorn et al. (2022) and Stotsky (2017). CA is based on the following principles:

- Educator designs the ILOs (Intended Learning Outcomes) using the verbs which describe the content to be learned.
- Educator creates the TLAs (Teaching and Learning Activities) including the assessment methods which address these verbs. Common verbs form the basis for constructive alignment of teaching and assessment.

All the components such as ILOs, TLAs and assessment tasks (as the part of learning) are based on the same verb model in the constructively aligned course. The results of instruction are massively improved when curriculum and assessment methods are aligned, due to the teaching quality assessment (Biggs, 1996). The literature and the difficulties associated with application of this concept to automatic control education are reviewed in the next section, which formes the basis for further development described as contributions of this paper in Section 3.

2. PREVIOUS WORK & MOTIVATION FOR DEVELOPMENT

Final written examinations is the most important part of summative assessment (evaluation of the student learning at the end of the course) and the last step in the constructive alignment associated with the course. Preparation of constructively aligned examination (which measures precisely the ILO competences) with a given number of points takes significant amount of time of the control educators, Das et al. (2021). This strongly motivates development of a toolkit for automatic compilation of examination problems, which facilitates constructive alignment. The toolkit should provide all necessary information for successful alignment of examination problems with ILOs, Amria et al. (2018).
To the best of our knowledge, this is the first work which proposes such a toolkit for automatic control courses. The closest work (presented by Grün et al., 2009) is associated with the development of automatic generation of examination problems in statistics using R.
An interesting literature review on automatic generation of examinations and the relation to Bloom's taxonomy can be found in (Ndirangu et al., 2021). Note that the educational

⋆ This work was not supported by any organization

10.1016/j.ifacol.2024.08.460

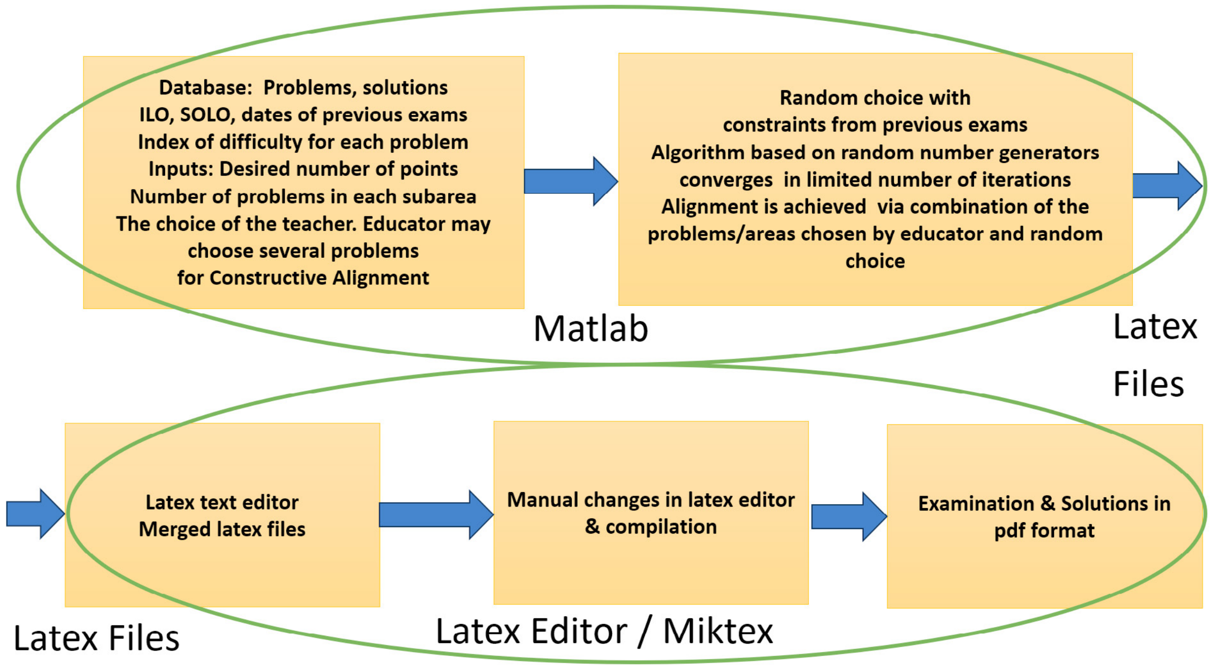

Figure 1. The flow chart for automatic generation of examination. This system consists of two large parts. The first part is written in Matlab and is associated with the database, which contains examination problems and solutions, learning outcomes, SOLO model, indices of difficulty and dates of previous examinations. The educator specifies the number of points and the number of problems in each area to be included in the examination. The educator may select several problems manually for CA. The Matlab program creates the LaTeX file by merging items of randomly selected problems in the subareas. The merged LaTeX file can be reviewed in WinEdt editor for example, where the educator gets opportunity to edit the examination and solutions. The LaTeX source file is compiled by Miktex which produces the pdf output with examination and solutions.

taxonomies, like Bloom's taxonomy were designed as generic methods supposed to be the same in all subjects. However, known taxonomies are not able to fit well to all subjects, which resulted in the development of so-called subject oriented taxonomies, for example the computer science-specific taxonomy developed by Fuller et al. (2007).

Unfortunately, the concept of CA is not directly applicable to the automatic control courses due to possible misinterpretation of the verbs that may appear in the verb based model, which in turn results in robustness problems. For example, the verb 'describe' can be associated with different levels of understanding. For example, the student can 'describe' simple relationships, or complicated cascade control systems which require evaluation and analysis. In this case different interpretations of the verb 'describe' should be recognizable by the system and overgeneralization should be avoided (Biggs & Collis, 2014).

These difficulties can be avoided via development of the SOLO (Structure of the Observed Learning Outcome) taxonomy associated with the subject of automatic control, and quantification of the ILOs using the developed SOLO model (see for example Stotsky, 2017). Proposed solutions are outlined in the next section.

3. PROPOSED SOLUTIONS & MAIN CONTRIBUTIONS

Each learning outcome and examination problem in the database is associated with the SOLO model, which was developed for the control course. Educator gets the opportunity to evaluate each examination item, (see Liotino et al., 2022) and to align the examination using the toolkit described in this paper. This approach is exemplified in Appendix, where the ILOs are quantified using the SOLO model for the course 'Process control and measurement techniques'.

The rapidity of the computer program (which is a measure of the software performance) is a very important requirement for this system since multiple runs may be required for CA. SCA (Stepwise Constructive Alignment) is a new concept introduced here and supported by the developed toolkit. The educator reviews randomly suggested choices of problems in each step of the alignment in terms of relation to ILOs, SOLO levels, levels of difficulty, etc., and selects the problems which are relevant for CA to the next run. The method allows to align the examination step-by-step, whereby the desired number of examination points is automatically guaranteed by the program in each run. SCA can be easily done with computer aided tools such as the decision support Matlab/LaTeX toolkit developed in this paper. The flow chart for SCA is presented in Figure 2.

4. THE CHOICE OF SOFTWARE & MOTIVATION FOR CUSTOMIZED DEVELOPMENT

The LaTeX typesetting system is far superior to Word, Open Office and other text editing programs due to (1) high typographical quality of the documents, (2) automatic formatting, itemization, indexing, equation numbering and reference generation, (3) flexible structures of the documents, (4) convenient typesetting of complicated mathematics, figures, tables and general technical contents, (5) a large and increasing number of free add-on packages which facilitate typesetting, (6) a number of customization options, which allows to create documents with user defined specifications, (7) compatibility with a number of operating systems, like Windows, Linux and MacOS, (8) support from a large number of communities like computer scientists, mathematicians, engineers and many others who represent both developers and users.

LaTeX is widely used in the control community for prepara-

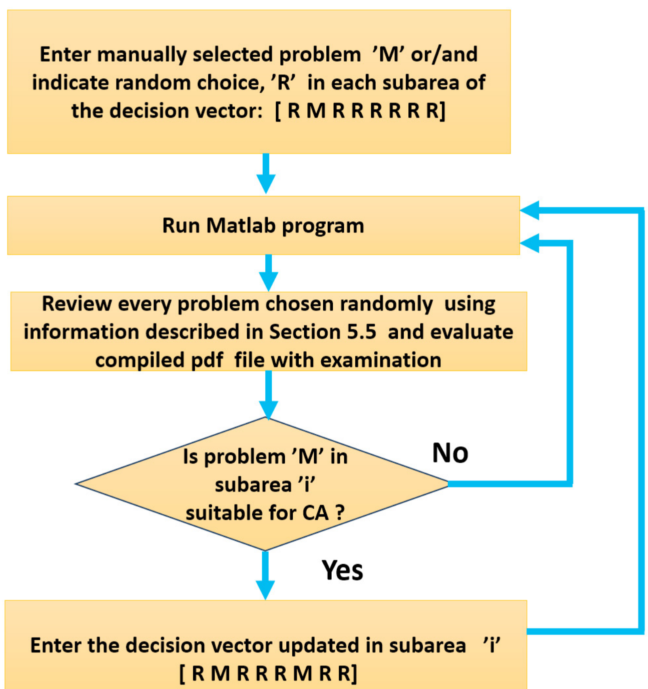

Figure 2. Flow chart for stepwise constructive alignment. The alignment starts with selection of an initial decision vector, where the educator enters any manually chosen problems (M) and indicates subareas where the problems should be chosen randomly (R). For example, the problem which has the number *M* is chosen manually for the second subarea of the decision vector, i.e. [*R M R R R R R R*], and all other problems in the seven other subareas are chosen randomly.
Educator runs the program, reviews information provided by the system in terms of ILOs, SOLO levels, indices of difficulty etc. (see Section 5.5), evaluates compiled pdf output file and makes the decision about the inclusion of the problems in examination by updating the decision vector. The educator runs the program again with the same decision vector, if suitable problems could not be found in the trial. The desired number of points is guaranteed by the system for each run and the system automatically rejects the problems which have been used recently. After a number of reruns the educator should be able to find the best possible solution for CA with appropriate indices of difficulty. The alignment is performed in a stepwise way where each step is associated with an update of the decision vector.

tion of collaborative research articles and other scientific documents. Also final written examinations are generally prepared in LaTeX at 'Division of Systems & Control' and a large number of examination problems written in LaTeX has been accumulated over the years. LaTeX was therefore chosen in this development as the software platform for preparation of the examination documents.
One additional software platform is needed for development of the database and performing advanced calculations required for the toolkit. Matlab is a high level programming language with flexible data structures, which are needed for development of the database. At the same time Matlab is a well developed computational platform that includes a large number of mathematical functions, which are needed for automatic selection of the examination problems. Matlab comes with a number of toolboxes designed for automatic control systems and has powerful visualization tools.
Many examination problems at the 'Division of Systems & Control' have been prepared and visualized using Matlab. Therefore, Matlab was chosen as the second platform for development of the toolkit. Matlab Database Toolbox could have been used for development of the database of examination items. However, number of limitations and the cost of this toolbox were the motivation for choosing a customized development instead. The database was associated with nested cell arrays, dynamic structure and classification codes which address unique requirements and facilitate automatic selection for the specific course. Note that nested solutions with dynamic memory allocation are not supported by SQLite or any other standard SQL database included in the Database Toolbox.
The rest of the paper is organized as follows. A description of the toolkit is presented in Section 5. Application of the developed system to the new concept of SCA, where the alignment is achieved by a number of software runs associated with random trials, is discussed in Section 6 (see also Figure 2). The paper ends with brief conclusions and outline for further development in Section 7.

5. DESCRIPTION OF THE TOOLKIT

An overview of the toolkit is presented in Figure 1, where selection of the examination problems is performed in Matlab and LaTeX is used for preparation of the examination documents. The description of the toolkit starts with the presentation of the structure of the output LaTeX file and examination items in Section 5.1. Classification codes and Matlab database are presented in Section 5.2. The algorithm for random selection of the examination items is described in Section 5.3. Input and output data for the system are presented in Section 5.4 and Section 5.5, respectively. Finally, the Matlab code for merging selected LaTeX items is described in Section 5.6.

5.1 Output LaTeX File

The output LaTeX file with the examination problems (composed in Matlab, see Section 5.6) is divided in three parts. The first part includes document style/class, packages, user defined commands and the commands which open the document, present front matter or heading and start enumeration/itemization. The second part contains examination items taken from the database using the algorithm described in Section 5.3. The examination problems are included in the document using the command `\item`. The third part closes enumeration and ends the document. Note that selection of the examination items occurs in the second part only whereas the first and the third parts are the same for all the trials, see also Section 5.6.

```
%     Part 1
\documentclass[12pt,a4paper]{article}
\usepackage{graphicx}
...
\begin{document}
% front matter or heading
...
\begin{enumerate}
%     Part 2
  \item first  problem
  \item second problem
  \item third  problem
  ...
%     Part 3
\end{enumerate}
\end{document}
```

The output LaTeX file can be edited and compiled using Miktex freeware which produces the pdf examination file.
Each examination item in the database begins with the command `\item` which makes the problem compatible with the main LaTeX document. For example, the item which is associated with the Control Design (see classification codes in Section 5.2) has the following form:

```
% Control design
% PI PID
\item The following process
$$
G(s)=\frac{1}{s^2+2s+2}
$$
is controlled with the PI-controller
$$
G_r(s)=K(1+\frac{1}{sT_i}),
\qquad K>0,\: T_i>0
$$
For which values of $K$ and  $T_i$ is
the closed loop system stable ?
\points{4}
```

which also contains the number of points determined by the user-defined command `\points{'number of points'}`. A comprehensive source of problems written in LaTeX and used in the previous examinations was available at the 'Division of Systems & Control'.

5.2 Database of Examination Items & Classification Codes

The database of examination items was developed in Matlab for the course 'Process control and measurement techniques'. The following classification codes associated with main subareas of the course were created for efficient storage and retrieval:

(1) Laplace Transformation
(2) Modeling
(3) Analysis
(4) Control Design
(5) Control Structures
(6) Electrical Systems
(7) Sampling & Filtering
(8) Observers

Each subarea is, in turn, divided into several topics. An overview of this course and ILOs are presented in Appendix.
The development of the system started with creation of the database, structured as eight Matlab files associated with the classification codes. The Matlab database is based on nested cell arrays which may contain any types of data (numbers, characters, strings and others) that are easily accessible by numeric indices, which facilitates automatic selection.
Each problem in every subarea is enumerated and associated with

- classification code (subarea and topic)
- number of points
- index of difficulty
- date (month and year) when the item was used last time
- path to item
- ILO
- SOLO model

Having the database organized in this way allows to display information relevant for CA for the automatically chosen problem (see Section 5.5). Note that the toolkit has been developed as a decision support system, and all the decisions about CA are made by the educator.
Note also that the index of difficulty from one to five (where $1-2$ is easy, 3 is moderate and $4-5$ is hard) was assigned to each item. The index of difficulty is associated with the ratio of the number of correct responses divided by the total number of responses, which can be determined using statistical data analysis of previous examinations (see for example the methods proposed by Johari et al., 2011 and Perez et. al, 2012). This index may change with time and thus may require updating.
Note that the index of difficulty can also be associated with cognitive load placed on the student by the problem (see Sweller, 1988) and with a 'SOLO average' (see Brabrand & Dahl, 2008).
The index of difficulty of examination (total index of difficulty) is associated with the sum of indices of individual items. Note that automatic generation facilitates preparation of equally difficult examinations (which have approximately the same total index of difficulty and/or desired distribution of indices) every time.

5.3 Selection of the Items

Selection of the examination problems is easily conducted in Matlab using random number generators so that the sum $\sum_{i=1}^{N} P_i$ converges to the desired number of points N_{points} (for example $N_{\text{points}} = 30$):

$$\sum_{i=1}^{N} P_i \underbrace{\rightarrow}_{\text{random trials}} N_{\text{points}}$$

where P_i is the number of points of each problem in subarea i, and N is the number of subareas.
Note that the algorithm automatically rejects the problems which have been used recently (a constraint specified by the educator) to avoid overlaps with recent examinations.

5.4 Input Data

The educator specifies the desired number of points for examination and the algorithm, described in Section 5.3, converges after only a few iterations. The system offers the possibility to choose specific problem manually in any of the subareas. In fact, the educator can manually choose all the problems in the examination. There is also the possibility to exclude problems from any of the eight subareas of the course described in Section 5.2. At the same time there is the possibility to include two or more problems in the same subareas.

5.5 Information in the Decision Support System

The following information is displayed to the educator for each automatically selected problem after each run of the program:

- Subarea of the course.
  For example: Modelling.
- Topic in this subarea.
  For example: Physical Modelling.
- Problem number, number of points, index of difficulty, the date (month and year) when it was last used.
  For example: Problem 2, 5 points, Difficulty 4, Sep 20.
- The path to the item.
  For example: Modelling/Mod003/001/filename.tex

- Learning outcome.
  For example: to develop mathematical models of technical systems and chemical processes based on first principles.
- SOLO levels.
  For example: SOLO 4: create and develop model, apply first principles.

The educator gets information about the topic in the subarea, problem number (which can be selected for the next run), learning outcome for CA, index of difficulty and SOLO model (which describes what the student will do in this problem). This information can then be used for SCA, see Section 6.

5.6 Merging Selected LaTeX Items in Matlab

The MATLAB program merges examination items selected by the algorithm described in Section 5.3 into one LaTeX output file as follows:

```
merge =[fileread('part1.tex')] ;
for i = 1:N
 merge =[merge fileread('item i')] ;
end ;
merge = [merge fileread('part3.tex')] ;
file = fopen('exam.tex','w')     ;
fprintf(file,'%s',merge)    ;
fclose(file)   ;
```

where the files 'part1.tex' and 'part3.tex' represent Part 1 and Part 3 of the LaTeX program presented in Section 5.1, respectively. Part 2 includes *N* appended items. The final pdf file with examination is produced by compiling the LaTeX file 'exam.tex'.
Note that the pdf output can be produced directly in Matlab using the function 'system(command)' which calls the operating system to execute the specified 'command'. The pdf output can then be produced directly by the command 'pdflatex'. This method is not recommended since the educator does not review the LaTeX file. And then also, operation waits for the 'command' to finish execution and errors in the LaTeX file may then result in freezing.

6. THE TOOLKIT AS DECISION SUPPORT SYSTEM FOR STEPWISE CONSTRUCTIVE ALIGNMENT

When preparing the final examination the educator evaluates all the TLAs including formative assessments which have been done in the course, and uses the possibilities offered by the system to align the final examination. For example, the educator may exclude problems from subareas that have been assessed by other methods, like exercises, laboratory works, formative tests and others. At the same time, the educator may include several problems in subareas which were not tested properly during the course. The opportunity to include manually any problem from the database is especially beneficial for CA. This option can be called CT (Choice of Teacher). The educator is able to choose several problems relevant for CA in some subareas and let the system randomly choose the rest of examination. For example, the educator may update the database and include new problems in the examination that are especially relevant for the current course.
Note that each examination problem in the database is associated with the ILOs and quantified using SOLO verb model and the index of difficulty, see Section 5.5. Automatic generation of examination for a given number of points takes a few seconds in Matlab and the educator may run the program several times in order to choose the best solution. The educator may include the problems, which randomly appeared in the previous run to the next run according to the concept of SCA (Stepwise Constructive Alignment). In other words the educator reviews the random choice of problems in every subarea in each step, and manually selects the problems to be kept for the next run, introducing additional constraints in the next run. The flow chart for SCA is presented in Figure 2.

7. CONCLUSIONS & OUTLOOK

Significant improvements of the results of learning due to application of the concept of CA and development of modern software tools (like Matlab & LaTeX) was the motivation for developing a toolkit for automatic compilation of examination problems in automatic control courses. The decision support Matlab/LaTeX toolkit, based on random number generators for selection of examination problems, formed the basis for development of a new strategy, Stepwise Constructive Alignment, which has the potential for further enhancement (compared to traditional CA) of the learning.
Further improvement of alignment can be associated with software development and introduction of ILOs, SOLO and indices of difficulty in the selection algorithm to automatically reject bad trials, choose desired level of difficulty, facilitate SCA and provide additional support to the educator. These improvements may result in development of a new concept of Automatic Constructive Alignment based on computer aided technologies which would allow to reduce the involvement of skilled educator/instructor (that implies significant time savings) in preparation of educational materials.
Further development can also be associated with updating parameters in the examination problems using toolboxes (like control system toolbox, filter design toolbox, signal processing toolbox, symbolic math toolbox, system identification toolbox, robust control toolbox, and many other toolboxes) which are available in Matlab.

AUTHOR CONTRIBUTION STATEMENT

The authors confirm contribution to the paper as follows:
Torsten Wik: conceptualization (lead), providing course materials, classification codes, data and problems in LaTeX , reviewing, editing.
Alexander Stotsky: conceptualization (SCA, SOLO), software (programming & database development), draft preparation, editing.
The authors regularly discussed challenges and improvements in the system, reviewed the results and approved the final version of the manuscript.

ACKNOWLEDGEMENT

The authors are grateful to their colleagues at 'Division of Systems & Control' for interesting discussions of an early version of this work.

REFERENCES

Amria, A., Ewais, A. & Hodrob, R., A Framework for Automatic Exam Generation based on Intended Learning Outcomes, In Proceedings of the 10-th International Conference on Computer Supported Education (CSEDU 2018), pp. 474-480.

Biggs, J., Enhancing Teaching Through Constructive Alignment, Higher Education, vol.32, N 1, 1996, pp. 1-18.

Biggs J. & Collis K., Evaluating the Quality of Learning: The SOLO Taxonomy (Structure of the Observed Learning Outcome), Academic Press,2014.

Brabrand C. & Dahl B., Constructive Alignment and the SOLO Taxonomy: A Comparative Study of University Competences in Computer Science vs. Mathematics, Conferences in Research and Practice in Information Technology, vol. 88, 2008, pp. 3-17.

Das B., Majumder M., Phadikar S. & Sekh A., Automatic Question Generation and Answer Assessment: A Survey, Research and Practice in Technology Enhanced Learning, vol. 16, N 5, 2021, pp. 1-15.

Fuller U., Riedesel C., Thompson, E., Johnson C., Ahoniemi T., Cukierman D., Hernan-Losada I., Jackova J., Lahtinen E., Lewis T. & Thompson D., Developing a Computer Science-Specific Learning Taxonomy, ACM SIGCSE Bulletin, vol. 39, N 4, 2007, pp. 152-170.

Grün B. & Zeileis A., Automatic Generation of Exams in R, Journal of Stat. Software, vol. 29, N 10, 2009, pp. 1-14.

Johari J., Sahari J., Wahab D., Abdullah S., Abdullah S., Omar M. & Muhamad, N., Difficulty Index of Examinations and Their Relation to the Achievement of Programme Outcomes, Procedia - Social and Behavioral Sciences, vol. 18, 2011, pp.71-80.

Knorn S., Topalovic D. & Varagnolo D., Redesigning a Classic Control Course Using Constructive Alignment, Student Centred Teaching and Continuous Assessment, IFAC PapersOnLine 55-17, 2022, pp. 180-185.

Liotino M., Garone A., Knorn S., Varagnolo D., Garone E. & Fedeli M., Assessing Engineering Exercises. A Novel Taxonomy, IFAC PapersOnLine 55-17, 2022, pp. 49-54.

Ndirangu P. & Muuro E., A Literature Review on Automatic Generation of Examinations, Open Journal for Information Technology, vol. 4, N2, 2021, pp. 77-84.

Perez E. , Santos L., Perez M., Fernandez J. & Martin R., Automatic Classification of Question Difficulty Level: Teachers' Estimation vs. Students' Perception, Frontiers in Education (FIE) Conference, Washington, DC, United States, Oct. 3 - 6, 2012, pp. 1 - 5.

Stotsky A., Modified SOLO Taxonomy Model for Constructive Alignment in Automatic Control & Signal Processing Education, Report in Chalmers Publication Library, Gothenburg, Sweden, 2017, https://research.chalmers.se/publication/249992

Sweller J., Cognitive Load During Problem Solving: Effects on Learning, Cognitive Science, vol. 12, N2, 1988, pp. 257-285.

APPENDIX: ILOS & SOLO MODEL FOR THE COURSE 'PROCESS CONTROL AND MEASUREMENT TECHNIQUES'

The course 'Process control and measurement techniques' ('Regler- och mätteknik') is given at Chalmers University of Technology as a basic course on dynamics and control of linear systems corresponding to 6 ECTS (European Credit Transfer System) credit points. The examination, which gives 4.5 credits, is an important part of summative assessment of the course. The course covers the following topics: (1) Development of models, based on physics or experimental data for linear and non-linear systems, (2) Linearization of state equations and derivation of transfer functions, (3) Analysis of linear dynamic systems. Analysis of feedback systems, (4) The Nyquist stability criterion, (5) P, I, PI, PD, PID and PIPD controllers, and their main characteristics, (6) Bode diagram. Non-minimum phase system. Design of control systems in the frequency domain, (7) Sensitivity functions and robustness, (8) Control structures. Feedforward, cascade and Otto-Smith control, (9) Computer control, sampling and discretization, (10) Sensors, analog and digital filters, (11) Simple DC and AC circuits. Bridge structures and connections.

ILOs for the course can be formulated as follows. After the course 'Process control and measurement techniques' the student will be able:

- to describe and explain basic concepts of control theory, SOLO 2 (describe, understand definitions, explain basic concepts).
- to use block diagrams, transfer functions, state-space representations (and relations between those) for description of control systems in different applications, SOLO 2 (describe structures, part/whole/nested, apply simple procedures, simplify, understand relations, solve simple equations, make use of experiments/measurements).
- to explain concepts and applications of the control system stability, associated with the stability of differential equations, SOLO 3 (explain concepts, analyse concepts), SOLO 4 (evaluate using mathematical methods, do algorithm).
- to quantify the performance of open and closed loop systems using impulse response, step and frequency responses, SOLO 2 (make use of measurements), SOLO 3 (identify parameters), SOLO 4 (evaluate using mathematical methods, analyse performance).
- to design feedback and feedforward control strategies which provide desired properties to the closed loop systems, SOLO 4 (create and develop structures, design feedback/feedforward, elaborate, estimate variables/parameters, test theory, modify, improve).
- to quantify the performance of closed loop systems using time domain and frequency domain methods, SOLO 4 (evaluate, examine, test theory, compare, improve).
- to develop mathematical models of technical systems and chemical processes based on the first principles, SOLO 2 (describe basic relations), SOLO 4 (create and develop models, compose, estimate/evaluate, apply first principles, apply energy, force and material balance).
- to understand and apply basic principles of sensing and measurements, filtering (including RC and Kalman filtering), state estimation (in form of observers) and signal processing methods, SOLO 2 (describe, understand definitions, compare), SOLO 3 (explain causes and concepts), SOLO 4, (apply first principles, estimate, filter, do algorithm).

Modified SOLO taxonomy model described by Stotsky (2017) was applied for quantification of the ILOs in the course 'Process control and measurement techniques'.