ISBN: 978-3-95450-214-1 ISSN: 2673-5490 doi:10.18429/JACoW-IPAC2021-MOPAB047

A CAD TOOL FOR LINEAR OPTICS DESIGN: A USE CASE APPROACH

J. Bengtsson[†], Helmholtz-Zentrum Berlin, BESSY, Berlin, Germany
W. Rogers, T. Nicholls, Diamond Light Source, Oxfordshire, U.K.

Abstract

The formula relevant for *linear optics design* of synchrotrons are derived systematically from first principles; an exercise in *Hamiltonian dynamics*. Equipped with these, the relevant *use cases* are then captured for a streamlined approach allowing professionals, i.e., *software engineers*, to efficiently prototype & architect a CAD tool, like those available to mechanical engineers since the mid-1960s. In other words, *robust design* of a modern synchrotron is an exercise in/pursuit of the art of *engineering-science*; i.e., *solving the inverse problem*: from end-user requirements to a conceptual & engineering design.

BACKGROUND

Nordic Contributions to Software Engineering

Computing has a long strong tradition in the Nordic countries, originating from the late 1940s, mainly as an engineering activity to build computing devices to assist mathematicians & engineers in solving scientific & engineering-science problems.

In the 1960s several Scandinavian technological institutes were in the forefront of computer language development [1]. Algol-60 was the first computer language defined by the *Backus-Nauer form* (BNF) for *Chomsky's context-free grammars* (J. Backus was a programming language designer at IBM & P. Nauer a prof. in software engineering at univ. of Copenhagen). After which Pascal & C followed suit. Fortran on the other hand is context sensitive. The Algol based simulation language *Simula* was implemented by O.-J. Dahl & K. Nygaard at the Norwegian Computing Center 1966 (which introduced *class* & *object* & influenced the design of Smalltalk, C++, Java, and C#).

First generations computers, such as SMIL, built at univ. of Lund, had an Algol-60 compiler provided by T. Ekman, 1962; guided by C.-E. Fröberg, an expert on numerical analysis. The hardware was based on J. von Neumann's IAS architecture.

The originators of Turbo Pascal (A. Hejlsberg) & C++ (B. Stroustrup) [2] hailed from Denmark. Similarly, the prof. at the dept of automatic control, K. Åström, Lund institute of technology (LTH) was an expert in his field [3].

These pioneers built solid software engineering departments. So as a TA at the dept for ditto at LTH, reporting to T. Ekman, the first author had scrutinised student assignments on structured programming, for scalable & predictable results, in Fortran, Algol, and Pascal. The latter was added to the curriculum at the time. To pursue R&D and *engineering-science* requires constantly learning new tools & techniques.

Observations

While spending a semester as a technical student at CERN the first author was baffled when learning that it took a colleague/grad student pursuing applied physics in the LEAR group (*Low Energy Antiproton storage Ring*), a 1-month effort to isolate a bug in MAD-6 [4]. I.e., as part of an attempt by the group to use it for modelling of LEAR. A "small ring" storing antiprotrons (80 m circumference), enabled by *stochastic cooling* [5]. Her printout – a 6" pile of ~40,000 lines of Fortran code – which was not organised as a software library, used some ad hoc *MAD input language* [6]; i.e., mixing the lattice description/definition with the commands for the actions on it in the input file. For a perspective, Simula, mentioned in the previous paragraph was a programming language/tool for *discrete event simulations*.

For a streamlined approach: just "script"/code the simulations in e.g. Fortran-77 – the first line being a function call to read in the lattice file – followed by the commands/function calls, compile it, and then link to a *beam dynamics library*. The "script" could then also contain/provide e.g. function prototypes for new algorithms as well which, eventually, after testing, are ported to the *beam dynamics library*; providing a *recursive strategy/process for successive refinements*.

Contrarily, MAD-6 was essentially, a Hodgepodge/monolith of lifted modules/codes [7-11] developed at other government funded labs. Each had somewhat different approximations for the equations of motion/Hamiltonians & and integration methods. It is no surprise to years later read that [12]:

> *Inspection of the original MAD-X (MAD-8) code revealed that the Conte-Martini formulae [6] were the ones implemented (presumably they had been copied from ZAP [7]), and not the original expressions from Bjorken and Mtingwa [5]. Correcting the formulae for the horizontal plane and adding the terms containing vertical dispersion resulted in the new MAD-X module.*

For clarity, the confusion regarding the actual simplified equations of motion and "black box" approach to the model/code refers to MAD-8. Not surprisingly, it had to be re-factored & overhauled for the LHC design & commissioning.

Later an opportunity emerged to pursue the former, i.e., *first principles* approach, when providing and validating an on-line model for the Advanced Light Source commissioning (1990).

Prior to that, when at CERN as a scientific fellow – for a model to improve the control of the nonlinear dynamics

† johan.bengtsson@helmholtz-berlin.de

for LEAR to extend its low energy range – the first author chose DIMAD [13] (a symplectic version of DIMAT [14]); i.e., ~5,000 lines of Fortran code. I.e., a model/code that he could modify as needed for the task(s) at hand.

It was also used for an on-line model by interfacing to the control system with a macro assembly module for the I/O & a Pascal module for conversion from *engineering* to *physics units*; akin to a modern *Middle Layer* [15]. Straightforward, since the control system was implemented as a *client/server architecture* & *relational database* [16].

INTRODUCTION

Engineering-Science

Accelerator design is a pursuit of *engineering-science*, roughly, "neither science nor engineering", e.g., ref. [17]. Hence, because accelerator physicists tend not to have a strong background in an engineering discipline & related methodology, perhaps, it has not reaped the potential benefits from that approach.

In particular, one may compare with the evolution of the digital computer, aerospace [18, 19], and modern telecom [20].

One exception is MAX IV; i.e, MAX III <- II <- I. A *paradigm shift* in this field; achieved by introducing *disruptive technology(ies)* to enable miniaturisation to purse the $1/n^3$ reduction of horizontal emittance by increasing the number of dipoles n; vs. following the beaten path; pushing towards the *theoretical minimum emittance cell.*

SOFTWARE ENGINEERING

When the SLS project was funded – having completed the conceptual design, the first author joined a telecom start-up in Silicon Valley as systems engineer (for a telecom. system comprising ~1M lines of source code in C & C++ in a software development group with a staff of ~40).

For project management in the software industry there are concepts like: *spiral* & *agile model* (vs. *waterfall*), *use case approach*, *end-to-end testing*, etc. Similarly, mechanical engineers introduced the *finite element method* [21] & *CAD systems* in the 1960s. Other effective areas of standardization include ASME's *boiler and pressure vessel code*: conceived 1911 & first edition published 1915. And IEEE's crucial work for the emergence of modern telecom.

Another paradigm helping to push the state-of-the-art software is the *open-source* approach; e.g. pursued for *TeX* by D. Knuth first released 1978. A *typesetting system* vs. a *word processor* Knuth implemented to improve the printing quality for the computer science books he was writing [22]. For quality assurance he also provided the incentive [23]:

> *TeX is guaranteed to be bug free. The author, Stanford Professor Donald Knuth, will send you a reward check is you find a bug. The reward is currently $327.68 (that is, 2^15 cents).*

Needless to say, Unix & Linux have provided paradigm shifts in this field.

Use Case Approach

The *use case* approach is a strategy in system engineering developed by the software industry to capture the *functional requirements* from often vaguely defined, unknown, and changing needs for the *end user* of a complex system. In particular, by an intuitive approach where the *users* & *system* are visualized as a set of *actors* (end-users or other systems) interacting with a "black-box", see Fig. 1.

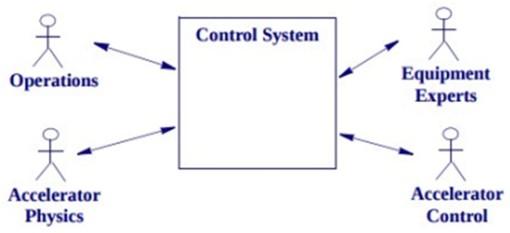

Figure 1: Use Case Approach (from ref. [24]).

It emerged as the solution to an intricate commercial *systems engineering* problem in the 1980s. To quote I. Jacobson, 2010, one of the "Three Amigos" who with G. Booch & R. Rambaugh invented & developed the *Unified Modeling Language* (UML) for software engineering in the 1990s [25]:

> *It was like that in the late 1960s and the '70s when the Ericsson AXE system beat all competition and won every contract thanks to being component-based. Similarly, when Rational was successful because of UML and Objectory. And Telelogic because of SDL.... In 1986, Use Case was the solution to the problem that traditional functional specifications were immense and not testable. To start from the users and find their different use cases made the specifications understandable, while we also at the same time found the test cases. The result was a good way to do test-driven development, now being popular in agile teams.*

He has since used the use case approach for professional collaborations with colleagues in the field [24, 26-28].

(On-Line Model -> Conceptual Design)^n

As mentioned earlier an opportunity arose to provide an on-line model – using a *first principles* approach for a self-consistent model (i.e., using the same equations of motion/Hamiltonian & integration method for both tracking & analysis of the global properties for the magnetic lattice, e.g., the linear optics) – to guide the Advanced Light Source commissioning (1990). In particular, it was:

- coded in Pascal as a *beam dynamics library*,
- "scripting" for the simulations was done in Pascal as well; for convenience & a recursive approach,
- hooked up to the *real-time database* (RT-DB) by *remote procedure calls*,
- the definitions for the magnet families in the lattice description file were extended to include a list of the RT-DB names for the corresponding power supplies

the latter two items providing a straightforward middle layer [15].

ISBN: 978-3-95450-214-1 ISSN: 2673-5490 doi:10.18429/JACoW-IPAC2021-MOPAB047

Since the underlying beam dynamics model had been validated by beam studies during the commissioning [29], the first author subsequently used it to guide the control of the nonlinear dynamics for the SLS conceptual design [30]. After which his successor re-used the beam dynamics model & related correction algorithms as a *model-server* for the commissioning of the system [31]. In particular, by machine translating the Pascal -> C (via p2c) and implementing it as a *Corba compliant client/server* in close collaboration with a software engineer.

Similarly, the C translation (open source) was utilised by the controls group for DIAMOND to implement a virtual accelerator – interfaced to EPICS as a *Virtual Input Output Controller (VIOC)*, see Fig. 2 – for e.g. end-to-end testing of controls applications; before the commissioning of the system [32].

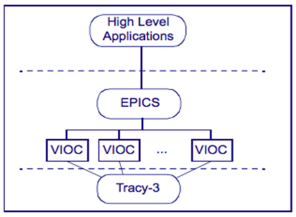

Figure 2: Client/Server Architecture (from ref. [33]).

Also, a manual translation of a subset of the core "numerical engine" from Pascal to C was made and then integrated with MATLAB to implement what's known as *Accelerator Toolbox* [34].

While the scope for EPICS since the inception has been "equipment control" [35]. For NSLS-II an opportunity arose to extend it to support model-based control by creating a software architecture to support model services in general [27, 28]. Hence, the computer model & related controls algorithms – the same as provided for ALS & SLS but now machine translated to C (courtesy open-source approach) – which had been used to guide the control of the nonlinear dynamics for robust design for the NSLS-II CDR & PDR, was available before the commissioning. Also, beam studies shifts were prepared by scripting Python notebooks beforehand to test/debug/validate them against the corresponding Virtual Accelerator [26]; to avoid wasting precious beam studies time. -> do it right the first time.

EQUATIONS OF MOTION

Synchrotrons are modelled by the Lorentz force:

$$\frac{d\bar{p}}{dt} = q\bar{E} + \bar{v} \times \bar{B}$$

which can be solved numerically.

Alternatively, for a systematic analytic approach, the relativistic Hamiltonian for a charged particle in an external electromagnetic field is [36]:

$$H(\bar{x}, \bar{p}; t) = q\Phi + \sqrt{(\bar{p} - q\bar{A})^2 - m_0^2 c_0^2}$$

i.e., a velocity dependent potential. And by utilising a Lie series solution for the Poincaré map [37]:

$$\mathcal{M} = \mathcal{A}^{-1} e^{:h:} \mathcal{R} \mathcal{A}$$

one obtains a recursive formulation; i.e., which can be automated by a *Turing machine*., e.g. a computer algebra system [36, 37].

By *linearising* one obtains the transfer/transport matrix, i.e., the *state matrix* for controls engineers, and a straightforward application of the *linear control theory* provides what's known as Courant & Snyder theory in this field. After which the methodology can be utilised for a systematic approach for feedback system design; e.g. *pole placement*, etc., see Figs. 3-4 & ref. [33, 38]. Similarly, the *action-angle coordinates* for Hamiltonian dynamics are referred to as the Courant & Snyder invariant & betatron phase. In other words, the beam dynamics model for linear optics design is a straightforward exercise in Hamiltonian dynamics. For the details see ref. [39].

As for the original development of bunch-by-bunch feedback systems, a technology choice had to be made: analogue or digital. As always, for a "novel" technology (often adopted from other fields): a risk for some [40], an opportunity for others [41]. Today digital off-the-shelf systems can be purchased from a small business; spin-off from a PhD thesis [42].

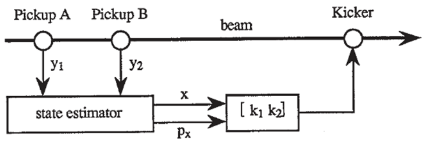

Figure 3: Linear Control Theory – State Space Approach (from ref. [38]).

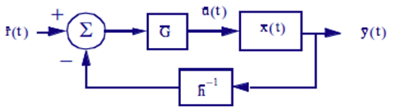

Figure 4: Linear Control Theory Methodology – Closed-Loop Control Paradigm (from ref. [33]).

MISSED OPPORTUNITIES?

We have applied the use case approach originating from the telecom software industry to a CAD tool pilot-project for linear optics design (see poster). Pointing out that such tools were introduced by mechanical engineers in the 1960s; and how the engineering process & methodology are integral to modern technologies, one may ask: Would today's synchrotrons look different with comparable tools for linear optics design?

ACKNOWLEDGEMENT

Although it has been a while the first author would like to acknowledge the advice given by Gottfried Mülhaupt – the driving force for the SLS in tunnel booster – when the Swiss Light Source project got funded and he was moving on to the private sector: *There is going to be pressure on you to cut corners. Resist that!*
Interestingly, Gottfried built & flew his own ultralight aircraft. For that one only gets one chance to do it right.

ISBN: 978-3-95450-214-1 ISSN: 2673-5490 doi:10.18429/JACoW-IPAC2021-MOPAB047

REFERENCES

[1] P. Sestoft, "Early nordic compilers and autocodes", in *Proc. HiNC'14*, May 2015, pp. 350-366.

[2] B. Stroustrup, *The C++ programming language*, Addison Wesley, 1986.

[3] K. Åström and T. Hägglund, *PID controllers: theory, design, and tuning*. Int'l Society for Measurement and Control, 1995.

[4] C. Iselin and J. Niederer, "The MAD program: methodical accelerator design; version 6, user's reference manual", CERN, Geneva, Switzerland, CERN-LEP-TH-87-33, 1987.

[5] S. van der Meer, "Stochastic cooling and the accumulation of antiprotons", Nobel Lecture, Dec. 1984.

[6] D. Carey and F. Iselin, "A standard input language for particle beam and accelerator computer programs", CERN, Geneva, Switzerland, CERN-LEP-TH-84-10, 1984.

[7] M. Zisman, S. Chattopadhyay, and J. Bisognano, "ZAP user's manual", LBL, NY, USA, Rep. LBL-21270 UC-28, Dec. 1986.

[8] A. Dragt, L. Healy, F. Neri, R. Ryne, E. Forest, and D. Douglas, "MARYLIE 3.0: a program for non linear analysis of accelerator and beam line lattices", in *Proc. PAC'85*, May 1985, pp. 2311-2313.

[9] M. Donald and D. Schofield, "A user's guide to the HARMON program", CERN, Rep. LEP Note 420, 1982.

[10] K. Brown, D. Carey, C. Iselin, and F. Rothacker, "TRANSPORT a computer program for designing charged particle beam transport systems", SLAC, CA, USA, Rep. SLAC-91REV1, 1974.

[11] K. Brown and S. Howry, "Transport/360: a computer program for designing charged particle beam transport systems", SLAC, CA, USA, Rep. SLAC-91, Jan. 1970.

[12] F. Zimmermann, "Refined models of intrabeam scattering", in *Proc. HB'06*, 2006, paper WEBY4, pp. 265-267.

[13] D. Douglas, E. Forest, and R. Servranckx, "A method to render second order beam optics programs symplectic", in *Proc. PAC'85*, May 1985, pp. 2279-2282.

[14] R. Servranckx and K. Brown, "Users guide to the program DIMAT", SLAC, CA, USA, Rep. SLAC Report 270 UC-28, 1984.

[15] G. Portmann, J. Corbett, and A. Terebilo, "An accelerator control middle layer using MATLAB", in *Proc. PAC'05*, May 2005, paper FPAT077, pp. 4009-4011.

[16] M. Chanel and T. Pettersson, "The LEAR control system - phase II", *Nucl. Instr. Meth. A*, vol. 247, pp. 146-152, 1986. doi:10.1016/0168-9002(89)90392-6

[17] S. Corneliussen, "The transonic wind tunnel and the NACA technical culture", NASA, Ch. 4, Rep. NASA SP-4119, 1988.

[18] P. Ceruzzi, *Beyond the limits: flight enters the computer age*. MIT Press, 1989.

[19] G. Padfield and B. Lawrence "The birth of flight control: an engineering analysis of the Wright Brothers' 1902 glider", *Aeron. J.*, vol. 107, pp. 697-718, 2003.

[20] A. Russell "OSI: The internet that wasn't" *IEEE Spectrum*, Jul 2013.

[21] R. Clough, "The finite element method after twenty-five years: A personal view", *Comput. Struct.*, vol. 12, pp. 361-370, Oct. 1980. doi:10.1016/0045-7949(80)90113-3

[22] A. Gaudeul, "Do open source developers respond to competition? the LATEX case study", *Review of Network Economics*, vol. 6, p.1, 2007. doi:10.2202/1446-9022.1119

[23] Ten Reasons Why TeX is Better than Word, http://web.mit.edu/klund/www/urk/texvword.html

[24] J. Bengtsson and M. Davidsaver, "An accelerator physics - software engineering collaboration: on-line model for FRIB commissioning", presented at *EPICS 2016 Collaboration Meeting*, ESS, Lund, Sweden.

[25] I. Jacobson, M. Christerson, P. Jonsson, and G. Övergaard, *Object-Oriented Software Engineering: A Use Case Driven Approach*, ACM Press, 1992.

[26] J. Bengtsson and Y. Hidaka "NSLS-II: Turn-by-turn bpm data analysis – a use case approach", BNL, NY, USA, Rep. NSLS-II ASD-TN-125, 2014.

[27] G. Shen and M. Kraimer "Server development for NSLS-II physics applications and performance analysis" in *Proc. PAC'11*, Mar.-Apr. 2011, paper MOP252, pp. 585-587.

[28] J. Bengtsson, B. Dalesio, T. Shaftan, and T. Tanabe "NSLS-II: Model based control - a use case approach", BNL, NY, USA, Rep. NSLS-II Tech Note 51, 2008.

[29] J. Bengtsson and M. Meddahi, "Modeling of beam dynamics and comparison with measurements for the advanced light source (ALS)", in *Proc. EPAC'94*, Jun.-Jul. 1994, pp. 1021-1024.

[30] J. Bengtsson, W. Joho, P. Marchand, G. Mülhaupt, L. Rivkin, and A. Streun, "Increasing the energy acceptance of high brightness synchrotron light storage rings", *Nucl. Instr. Meth. A*, vol. 404, pp. 237-247, 1998. doi:10.1016/S0168-9002(97)01168-6

[31] M. Böge and J. Chrin, "A CORBA based client-server model for beam dynamics applications at the SLS", in *Proc. ICALEPCS'99*, paper MC1P61, pp. 555-557.

[32] M. Heron *et al.*, "Progress on the implementation of the diamond control system", in *Proc. ICALEPCS'05*, Oct. 2005, paper P1_018.

[33] J. Bengtsson, "Design and control of ultra low emittance light sources", in *Proc. ICAP'09*, Aug.-Sep. 2009, paper TU3IOPK04, pp. 68-72.

[34] A. Terebilo, "Accelerator Toolbox for MATLAB", SLAC, CA, USA, Rep. SLAC-PUB-8732, 2001.

[35] M. Knott, D. Gurd, S. Lewis, and M. Thuot, "EPICS: A control system software co-development success story", *Nucl. Instr. Meth. A*, vol. 352, pp. 486-491, 1994. doi:10.1016/0168-9002(94)91577-6

[36] J. Bengtsson, "Non-linear transverse dynamics for storage rings with applications to the low-energy antiproton ring (LEAR) at CERN", CERN, Geneva, Switzerland, Rep. CERN-88-05, 1988.

[37] J. Bengtsson, "The sextupole scheme for the Swiss Light Source (SLS): an analytic approach", PSI, Villigen, Switzerland, Rep. SLS 9/97, 1997.

[38] D. Briggs, J. Bengtsson, and G. Portmann, "A linear control theory analysis of transverse coherent bunch instabilities feedback systems: the control theory approach to Hill's equation", Rep. CBP Tech Note-026, PEP-II AP Note 28-93, 1993.

[39] J. Bengtsson, W. Roger, and T. Nicholls, "A CAD tool for linear optics design – a controls engineer's geometric approach to Hill's equation", unpublished.

[40] W. Barry, G. Lambertson, and C. Lo, "Electronic systems for transverse coupled-bunch feedback in the advanced light source (ALS)", presented at *BIW'93*, Oct 1993.

[41] J. Fox, N. Eisen, H. Hindi, I. Linscott, G. Oxoby, and L. Sapozhnikov, "Feedback control of coupled-bunch instabilities", in *Proc. PAC'93*, Mar. 1993, pp. 2076-2081.

[42] D. Teytelman, "Architectures and algorithms for control and diagnostics of coupled-bunch instabilities in circular accelerators", thesis, Stanford Univ., SLAC, CA, USA, Rep. SLAC-Report-633, 2003.