STEP-NC驱动的智能数控技术及其对智能制造的现实意义

文/肖文磊　北京航空航天大学

数控技术的发展历史

世界上第一台计算机叫ABC计算机，是在1942年左右发明的。1952年MIT发明了世界上第一台数控机床，MIT在发明这套数控机床的同时，不仅发明了机床本身，同时定义了数字化制造的基础。最终形成了一种从计算机辅助设计、辅助制造到CNC的全过程流程，也就是至今一直在沿用的CAD-CAM-CNC制造模式。目前制造业本质上并没有颠覆这种制造模式，而是一直在做技术的改良。

今天，计算机革命使得很多行业得到长足发展。智能手机、智能汽车已广泛应用在生活中，人们自然希望把这些相关技术沿用到工业中，从而打造一个“工业4.0”架构。但是需要反思的是，所有的计算机相关技术其实都是“拿来主义”，比如说AI+NC，大数据+NC和CPS+NC等许多概念都是在计算机行业产生的革命，希望把计算机技术赋能到机床或是工业行业，从而产生智能制造的革命，但“拿来主义”能不能达成期望呢？从1942年发明第一台计算机，到1952年发明第一台数控机床，刚好差10年，从计算机发明开始，就一直在发生类似的故事。1995年，以太网的技术开始发展，2005年，工业以

太网技术也得到了大量的推广，成为现在的主流。回顾历史发现，工业技术发展一直在跟随着计算机行业发展，且刚好形成一套类似10年的周期规律，也就是说今天要做什么，要去回顾一下10年前在计算机行业发生的故事。

毋庸置疑，现在来看手机是智能的，但机床的智能化水平还不够，如何用计算机行业的智能水平赋能数控机床就成了热点。1940年左右，以计算机发明为代表。兼容化革命，在20世纪70年代左右，IBM斥资5.5亿美元用于突破软件的兼容性难题；在2010年左右实现了智能化，也就是智能手机终端的出现。那么数控机床要实现智能化，是不是也要分这三个阶段完成？到目前为止数控机床的兼容化还没有完成，在没有实现兼容化的前提之下，如何实现智能化呢？这是值得思考的问题。也就是说要回顾在历史上的兼容化难题是怎么解决的。谈到兼容化，必须谈到编程语言，数控编程语言的发展，从一开始的穿孔纸带到APT代码再到G代码，沿用至今。计算机编程语言，也是从穿孔纸带开始，发展到汇编、C++等各种语言，这两种编程语言的发展是极为不对称的，在计算机语言已经蓬勃发展的情况下，数控编程语言只发展了两代，会造成什么问题呢？要回顾计算机历史，当年是不是也有同样的困境？

G代码跟汇编语言类似，汇编语言在20世纪70年代广泛应用，汇编语言的问题就在于兼容性。为了解决这个难题，IBM斥资5.5亿美元，用来突破软件的兼容性难题。Frederick P.Brooks作为项目的负责人，获得了美国计算机领域最高声望的图灵奖。这个项目进展的同时，贝尔实验室开启了一个类似的项目Multics，也是为了打造一个操作系统，结果同样失败了。两个项目的失败，说明兼容性难题是不好解决的。项目的两个工程师，Dennis Ritchie和Ken Thompson决定开发一个操作系统，定了目标“小即是美”，成功开发了UNIX，顺便在操作系统的开发过程中发明了C语言。1 000个工程师失败了，但2个工程师成功了，这个问题后来被作叫软件危机，即落后的软件生产方式，无法满足快速增长的计算机软件需求。所谓的落后生产方式就是汇编语言的编程方式，已经没法满足快速增长的计算机软件的开发需求，从而导致软件开发和维护过程中出现了一系列问题。在历史上，计算机行业发生过多次软件危机，比较著名的是第一次软件危机，操作系统UNIX和C语言应运而生，解决了程序可移植且具有兼容性的问题。第二次软件危机归因于软件复杂度的进一步提升，发生在20世纪80年代到90年代，面向对象的编程语言（C++、C#和Java等）诞生了。说明软件危机是催生计算机革命的直接推手，产生了两个产物：一是操作系统，为了解决计算机硬件的兼容和软件可移植的问题；二是面向对象，为了解决大型软件的开发效率问题。

现在的G代码和数控系统水平或能级正好处在计算机编程语言汇编和定制操作系统的状态。G代码类似于汇编，是不可移植的。不同的机床要对数控系统做定制，也就类似于之前定制的超系统水平，计算机发展带来的启示，值得人们深思。

肖文磊

北京航空航天大学机械工程及自动化学院、航空高端装备智能制造工信部重点实验室副教授

1. 制造危机与智能制造

G代码沿用了五六十年了，但也有本质问题，例如信息缺失、不可追溯、不可交互且与上层系统不可兼容等，像漏斗一样，把大量的信息从CAD/CAM中漏除了。只剩下轨迹的执行指令，设计目的其实包含很多信息，经过后置处理后，就只剩下轨迹运动信息了，因为CNC只关注运动。在G代码的架构下会产生什么问题呢？会产生制造危机，即“落后的制造方式无法满足迅速变化的产品需求，从而导致产品设计与生产管理过程中出现一系列严重问题的现象”。这个想法与知名学者的想法不谋而合，华中科技大学杨叔子院士于1992年发表了一篇文章，文章中表示“人们对制造过程的自动化程度赋予了极大的研究热情，因为制造效率的提高非常快，几乎提高了20倍，但生产管理效率只提高了1.8～2倍，产品设计的效率只提高了1～2倍”“各种问题求解的最终决策在很大程度上仍依赖于人的智慧”“各种迹象表明现在正处于

制造历史上的一个危险时期”。危险时期即制造危机，直到今天，制造危机还在，因为绝大多数研究目前还停留在探讨人工智能在制造领域的应用阶段，近年来开发出了众多类型且水平各异的面向过程中的特定环节、特定问题的智能化孤岛，智能化孤岛对制造环境的全面智能化研究工作还处于起步阶段。

2. 数字化制造体系

在技术长足发展后，是不是只要扎根云制造概念，将CPS、数字孪生及AI等概念引入到制造环节，智能制造的本职工作就能达成呢？先进的制造工厂采取的数字化制造体系架构，是典型的“一横一纵”概念，所谓的“一横”是需求-设计-工艺-制造-服务环节，希望用软件或数字化的手段传递以取代人工传递方式。所谓“一纵”就是在制造环节，必须要跟机器打交道，会产生大量的软件，像ERP、MRP 和MES/MOM等。“一横一纵”各种软件形成数字化制造体系。可以简单地认为，是为了满足人需求的各种工业软件的集合，这些工业软件都有一个问题，就是一旦开发工业软件，就需要一个专业的人员进行操作，越专业的软件越需要专业的人，对人的智慧、脑力劳动要求很高。维护系统过程对脑力劳动依赖明显过高，这就是产生制造危机的本质原因，也是杨叔子院士文章中所写的问题。到今天为止，系统得到了长足的发展和不断的提升，但并没有解决减少人脑力劳动这一根本性问题。

3. STEP-NC兼容的制造模式

STEP-NC兼容的制造模式提出的目的就是期望把面向过程或基于G代码的逻辑改成面向对象的制造逻辑。期望通过迅速变化的客户要求，大大缩短产品的批量及全生命周期，CAD、CAM、CAPP和CNC之间需要一个平稳、高效、统一和开放的NC数据链，STEP-NC技术的出现实现了CNC系统与CAD/CAM系统之间数据信息的统一。STEP-NC也期望在CAD/CAM与CNC之间建立一个双向的数据流，从而让信息孤岛的现象得到缓解。这些概念都是在2000年提出的，但放到今天并没有过时，STEP-NC提出以后，世界上掀起了一阵研究热潮，在2006—2008年，全世界范围很多研究团队都在对相关的方向进行跟踪研究，但能够坚持下来的团队寥寥无几，坚持到今天形成系统，并有应用案例的可能有两家：一家是美国的STEP Tools；另外一家是北京航空航天大学的团队。

STEP-NC从2000年左右开始兴起，一直到2008年达到高峰，虽然研究热潮经历了一轮衰退，但任何研究背后都有Hype曲线规律，那么就要探究一下STEP-NC在相关研究中产生的问题，比如STEP-NC为何一直没有得到产业应用呢？STEP-NC的关注度为何由盛而衰呢？制约STEP-NC发展的瓶颈是什么呢？STEP-NC与智能制造之间的关系是什么？STEP-NC是不是智能数控必经之路？STEP-NC技术是否会应验否定之否定的历史发展规律？所谓否定之否定，就像Hype曲线一样，一个革命性的技术需要经历一个肯定、否定及否定之否定的三个阶段，得到螺旋上升。

一项具有革命性的技术，从概念提出到实际的应用，一般都需要30～35年，比如说世界上第一种面向对象的编程语言Simula67，这项发明早于操作系统的出现，就已经出现了面向对象的概念。当时面向对象概念并没有得到普及与发展，直到1998年出现了世界上第一门流行的面向对象语言C++，才开始得到普及应用。这个过程经历了32年，所以STEP-NC可能还有10～15年的发展过程。

STEP–NC驱动的智能数控技术

1. 制造仿真引擎GPCORE

根据计算机历史发展和数控历史发展，发现要解决软件危机或者制造危机，要做两件事：第一件事叫操作系统；第二件事叫面向对象。所以首先要打造面向制造的操作系统，制造的操作系统本质上是为了隔离制造的硬件。为了达到目的，开发了一个面向制造的仿真引擎GPCORE，采用了“小即是美”的概念，把各个模块分开，再动态组合，采用这个理念，分成各种模块进行开发，以隔离相应环节硬件。操作系统看起来能干的应用似乎很少，但如果没有操作系统，其他应用程序都将成为“无根之水”。提供一个操作系统的环境，使得其他程序和应用能够得到集成和可移植的开发。

根据这个理念，打造了GPCORE，分成模块化，即分为多层，有内核层、业务层及界面层，可以动态组合，内核互相独立。整套架构就形成制造仿真引擎，仿真引擎目前主要分为这几大模块，包括三维渲染引擎gpScene、STEP-NC建模引擎gpStepNc、铣削加工仿真引擎gpMilling、机床仿真引擎gpMachine及计算机仿真引擎gpRobot等，以满足不同领域智能

化应用需求。

三维渲染引擎专门打造了一个面向工业仿真的gpScene内核，主要的内核接口能够满足工业的常规需求，能够提供高效灵活以及功能强大的API接口；STEP-NC的建模引擎能够从使CATIA导出STEP-NC文件，开发了相应的插件，能把90%以上的信息导出，支持STEP-NC文件的读取与解析，可以根据硬件架构进一步自动透支出相应的执行代码；铣削加工仿真引擎主要是针对切割过程中在线仿真需求打造，采用了一个前瞻式体塑化切削过程，能够减少铣削仿真引擎时间，可以动态实时预测切削率；机床仿真引擎可以根据机床配置，实现机床的运动学和动力学仿真的效果；机器人仿真引擎可支持国内外主流机器人，灵活配置机器人尺寸，用户不需要具备机器人学的相关知识，就可以快速构建出机器人。

制造仿真引擎GPCORE典型的应用场景，目前正在研发的兼容STEP-NC的智能数控系统，致力于打造支持STEP-NC的CAD/CAM系统，目前发展到第三代，可以支持STEP-NC的输入，从而使信息能够无缝集成、无损传输。传统的数控机床短时间内不会淘汰，针对这种情况，开发边缘孪生数字孪生系统，可以采取OPC-UA或其他通信接口方式，把数控系统相关信息索取过来，起到实时仿真的目的，并且根据STEP-NC文件后处理生成G代码，达到了硬件屏蔽作用。机器人应用在线校准的应用场景，可以利用GPCORE内核中的gpRobot组件，提供对机器人运动学逆解等问题的强大支持，从而达到机器人在线监控和方向控制的目的。科德数控系统驱动虚拟机床运动仿真采用真实数控系统，通过接口驱动虚拟机床，从而实现在线调试，应用GPCORE可以很快速地搭建出原型系统。还有一个比较有意思的案例，利用GPCORE与北京大学合作，与生物领域学科交叉，开发了适应海量图像数据的大细胞分辨率智能追踪算法平台，从而追踪多个时间点目标单细胞发育的过程。这个算法取得了国际上前所未有的超高识别进度的效果。目前这套软件得到了进一步发展，在脑科学计划项目中，跟中科院自动化正在合作并应用于脑科学神经分割逻辑处理的过程。

2. 面向孪生的制造模式

时态的变化有别于数字样机和数字孪生，产生了静态绑定和动态绑定的概念。静态绑定是仿真的行为在运行前决定，而且在运行时不会变，在这个工业领域许多主流的仿真软件应用都在运行前。动态绑定的仿真行为在运行时决定，而且在运行时可以变，是在运行时同步仿真。时态产生了概念区别，回顾计算机历史，

C语言里有函数指针的概念，函数指针提供了动态绑定思想，函数的行为是在运行时决定，在运行时可以改变，在C++里动态绑定，最后发展成虚函数概念，后形成多态。多态是面向对象类的重要基础，最终驱动形成面向对象的概念。通过计算机历史，发现动态绑定是面向对象的基础，因此可以做一个大胆的假设，面向对象革命能否在智能制造领域再现？何为智能制造的面向对象，要怎么去实现？两个问题非常值得思考，所以就提出面向孪生的制造模式。

操作数控机床不仅要理解G代码，还要理解工艺本身是什么，基于此，致力打造能够实现单一责任和面向对象式的加工系统，采取兼容目前已有的CNC模式，打造一个边缘孪生系统，边缘孪生系统跟云孪生镜像进行呼应，每一个机床在云端都有ID。机床是一个实例化的过程，云孪生是一个抽象机床，在云端进行兼容化和批处理。只有抽象才能统一，这是面向对象思想的集合。

进一步思考，有大量工作需要去做，比如说面向对象的基石是继承、重载和多态，面向孪生制造的继承、重载和多态是什么样的呢？这个是值得深思的问题，值得未来打造。

小弗雷德里克 · 布鲁克斯在《人月神话》中提到“在未来10年内，无论是在技术还是管理方法上，都看不出任何突破性的进步，能够保证在10年内大幅度提高软件的生产率、可靠性和简洁性。”放到今天，制造业在未来10年内在技术和管理方面都看不出任何突破性的进步，能够保证10年内大幅度提高制造的生产率、可靠性和简洁性。到今天为止，面向对象确实是解决软件危机的重要武器，只不过要有面向对象的理念，用了面向对象的语言不一定写出来是面向对象的程序。推动面向发展的戴维 · 帕纳斯表示，面向对象跟各种复杂语言的联系紧密。可以讲数字孪生跟各种复杂的应用联系已经很紧密了，但人们并没有告诉面向对象是种设计方法，并向他们讲授设计的方法和原理，大家只是被告知面向对象是一种特殊的工具，结果人们使用这种语言做了不好的设计，从中没有获得多少价值，一旦获得的价值太少，就不会流行了。放在智能制造领域，数字孪生也许是一种制造模式，而不是一种特殊的工具。对于怎么去做下一步研究非常具有启示价值。

3. 面向智能数控加工的数字孪生系统

在数控加工中应用数字孪生技术，实现数控机床终端的智能化，分为边缘孪生和云孪生。首先可以在云孪生系统中，针对某个机床派发一个STEP-NC文件，预先没有指定具体机床，只要计算能够接收文件，文件就会被同步到边缘孪生系统当中。边缘孪生系统接收到文件，根据机床配置自动后置出来相应的G代码，进一步的DNC到数控系统当中，数控系统自动接送到加工文件，就可以直接运行，边缘孪生系统保证文件是没有任何问题，并且符合特定数控系统及机床的运行格式。文件运行后，相关数据自动被采集到边缘孪生系统中。边缘孪生系统根据采集到的过程数据，推理出来当前的工步，相关的关联信息都可以推理出来。推理的信息反馈到云端系统，云端系统采集相关的数据，可以推进相关的数据，进行实时监控和远程监控。数据同时存储到历史数据库中，便于未来追溯。同时开发了一个三维仿真系统，可以直观地看到机床的运动过程，三维仿真系统还有待发展，未来更多信息会集成到相关系统中去。系统也可以获得历史信息，所有信息都存在于云端系统，在发生质量或相关问题时，可以追溯过去。整个过程都是自动完成的，节省大量人工。

STEP–NC对智能制造的现实意义

1. 工业互联网与互联网的区别

工业互联网不等于工业+互联网，互联网主要解决的是人与人之间的连接问题，而工业互联网主要是解决机器和机器之间或者机器与人之间的问题，最终目标是要取代人或减少脑力劳动，但工作量并没有减少，要用软件来代替人去执行。工业互联网也得遵守发展10年周期率的影响，要符合工业的特性。只有减少人工，才能体现工业互联网的优势。所谓机器用的软件，要给机器赋予更大的权力。只有给机器赋予更大的权利，才能减轻人的脑力劳动。互联网主要解决的是人与人的沟通问题，工业互联网不但需要解决人和人的沟通问题，还需要解决人与机器、机器与机器工艺的沟通问题，跟原来的互联网存在一些本质的不同。

2. 云制造的现实意义

从需求角度探讨，一是从人的角度，打破机器的物理壁垒，人机交互更为便捷，用手机端就可以交互，生产过程更为透明，可以自动存储海量的数据；另一个

是从计算机角度或从机器角度，更重要的需求是把物理对象转化为变量。有PLC以后，可以把电动机映射成一个变量，只要操作这个变量就可以，把传感器映射成变量，将传感器和电动机的变量放在一起，就能用程序搭建出闭环控制系统。这就是所谓的化物理对象为变量，依赖于抽象而不依赖于具体实现的。

3. CPS应该怎么架构

目前很多CPS架构都不是真正的CPS架构，而叫Cyber-Agent-Physics，如果只是把数据传到云端，那么技术上是可行的，但缺少智能终端，只是采集数据而已，叫数据采集系统，只不过把基于物理模拟量的采集变成网线采集。要达到智能水平，必须有智能终端。边缘孪生软件只是一个过渡系统。

4. 数字主线的数据膨胀问题

在进一步开发应用的过程中，发现有一条数字总线，期望用数字总线技术把整个制造过程全部打通，这里存在一个很大的悖论，从“一横一纵”的发展态势来看，越往后信息量越大，对软件开发的压力巨大。如何解决数据膨胀问题，是数字主线遇到的主要问题，目前这个问题是制造STEP-NC和智能制造数据中心发展的主要矛盾。

5. 面向孪生的制造模式

需要构建孪生系统，孪生的镜像必须是抽象的镜像，不能是一个具体的镜像。如果是具体镜像，不同的镜像之间没法兼容，违背了面向对象的基本原则。所以构建面向孪生的制造模式需要采用类似理念，不需要关心这个机床多样化的的信息，更多关心加工能力、加工工艺及拓扑工作空间等，这样才能在云端形成统一的、成组且批处理的模型，构建V-Factory的制造模式，无论物理距离在哪儿，都能组成一个虚拟的加工车间。V-Factory可以打造出各种各样的生产模式，依靠不同的生产模式，应对不同产品设计的需求。

6. 智能制造核心目标

要打造一个从现有的五层架构转化为三层架构，包络物理层、孪生层和服务层，都需要相应的软件模块作为支撑才能实现，并且需要面向对象的理念进行架构。这个架构有别于数字化制造的体系，智能制造体系是满足机器需求的各种微服务的集合。

7. 更高维的软件危机

产生一个面向对象的根本性的问题，叫强耦合问题。每个节点都需要理解STEP-NC的数据全集。数据全集是非常复杂的，要求所有的服务都理解STEP-NC全集是不合理的。后面就有了相应的解决方案，比如Python是动态语言、Spring Boot做了依赖注入的概念。这些都是解决更高维软件危机的具体手段。借鉴这种理念，把它打造成面向动态的数据结构以及相应的编程依赖注入式的架构方式。当需要打造一个需要STEP-NC描述的系统时，可以进行降维操作，把图变成集合，但是数据还需要面向对象，所以就采用面向对象的数据库架构进行数据存储。面向对象数据库的语言是通过SDI技术自动创建的，在读取时，需要经历转换和裁剪的过程，一旦转换和裁剪面向具体用户，就会变得更为简单且更为降维，这就是动态的数据结构。采用动态技术结构可以让工作量直接降到原有的10%。既要有在内核层的面向对象，也要有在接口层的面向表格。数据采样在数字孪生中谈得比较多，目前机床的数据采样还是比较初级的，只是采样一些关键信号，仿造邮件、短信、电话和快递等沟通理念构造出一个边云协同的仿人通信模式，把机床的每个节点都想象成人的权利，存储的信息分别根据需求存在不同的数据库，分为实时、短时和长时的记忆。数据采样不是终极目的，边云协同才是。SCADA不是解决方案，仿人通信才是。原有的硬件架构不适合新型的智能制造模式，尤其是超系统和电子架构方面，机床需要下一代的智能机床硬件架构。

智能数控的“冰山理论”

智能数控体系的发展历程以及未来发展愿景，跟计算机行业发展非常类似，提出了智能数控的“冰山理论”，往往关注在“水面”以上的应用。“水面”以下还有大量的工作，需要构建一个系统，屏蔽的是G指令集，数控系统就变成了跟芯片一样的逻辑。再往上架构STEP-NC，STEP-NC面向对象以及其他语言，基于STEP-NC发展动态语言，连上CPS，从而连接到相应的微服务、智能服务，是不是也能形成类似的智能系统，这是未来值得去探讨的。EA

（本文源自《2022数字化车间和智能工厂》在线论坛第三期：布局工业通信和5G+工业互联网，编辑整理：周文静）

扫描二维码了解更多