

# 基于 RTLinux 的 AFM 实时反馈控制系统

周娴玮, 方勇纯, 董晓坤, 张玉东

(南开大学信息技术科学学院, 天津 300071)

**摘要:** 为了提高原子力显微镜(AFM)的测量性能, 利用 RTLinux 良好的实时性, 设计了一种开放式的反馈控制平台, 完成了基于 RTLinux 的 AFM 实时反馈控制系统的软硬件设计, 并在 PC 上具体实现这种实时反馈系统。测试结果表明, 该系统能较好地实现 AFM 在 Z 方向上的实时反馈控制。利用该平台良好的开放性, 可以将各种先进的控制方法应用于 AFM 系统中, 以提高原子力显微镜的扫描速度和精度。此外, 该基于 RTLinux 的 AFM 系统还可以扩展成为一个开放式的纳米操作平台。

**关键词:** 原子力显微镜; RTLinux 系统; 反馈控制

## Real-time Feedback Control System for AFM Based on RTLinux

ZHOU Xian-wei, FANG Yong-chun, DONG Xiao-kun, ZHANG Yu-dong

(College of Information Technical Science, Nankai University, Tianjin 300071)

**【Abstract】** Aiming at improving the measuring performance of Atomic Force Microscope(AFM), an open-structure controller is designed based on RTLinux with software and hardware constructed completely. The experimental results show that the RTLinux based control system is capable of controlling the AFM system. Thus more advanced control algorithms such as adaptive PID and  $H_{\infty}$  control are expected to be implemented on this open platform and to enhance the scanning speed and resolution. Besides, this control platform can be extended to an open nano-manipulation platform which will facilitate future research on nano-manipulation.

**【Key words】** Atomic Force Microscope(AFM); RTLinux system; feedback control

### 1 概述

自从 1986 年原子力显微镜问世以来, 它在各种与纳米技术相关的研究领域得到了广泛的应用, 人们形象地称它及相关技术为纳米科技的“眼”和“手”<sup>[1]</sup>。原子力显微镜的基本机理是利用扫描器驱动探针在被探测样品表面上进行扫描, 由于探针与被测样品之间微弱的相互作用力, 探针所连接的悬臂会发生微形变, 通过光检测系统观测这种微形变  $z$ , 并通过一定的反馈系统来驱动压电扫描器在 Z 方向上微动使得  $z$  保持为一设定值, 从而可以根据控制器的输出来获得物质表面纳米级的形貌信息<sup>[2]</sup>。

反馈系统是原子力显微镜的关键部分, 它直接关系着 AFM 扫描的速度和精度; 此外, 如果控制效果不好, 使光斑的示数  $z$  离设定值偏差较大, 则很容易导致针尖和样品之间的作用力过大, 从而毁坏样品和探针。当前的 AFM 系统一般都利用 PI 算法来进行反馈控制, 这种方法具有参数调节繁琐、带宽窄等缺点, 限制了 AFM 在快速生物反应过程上的应用<sup>[3-4]</sup>。因此, 近年来, 很多学者采用一些其他的先进控制算法, 如自适应 PID 控制、 $H_{\infty}$  鲁棒控制和自学习迭代控制算法等<sup>[3-5]</sup>, 来取代现有的 PI 控制算法, 以提升 AFM 系统的测量性能。遗憾的是, 由于商业机密方面的考虑, 当前的 AFM 控制系统开放性不强, 很难将这些先进控制算法在现成的 AFM 系统中直接进行应用。

### 2 系统总体结构设计

AFM 采用逐点扫描的方式, 为了提高系统的扫描速度, 控制系统必须具有使压电扫描器在不同样品上快速稳定的功能。对于通常采用的 1 Hz 的扫描速度, 如果图像分辨率为

512×512 像素, 则压电扫描器在每个点上的停留时间约为 1 ms。在这个时间内, 要求控制系统能使光斑稳定到设定值, 因此系统对于实时性的要求非常高。

一般的操作系统其本质上是分时的, 所以它们无法满足实时控制系统的硬实时要求。而商业的实时操作系统如 Vxworks, QNX 价格都非常昂贵, 而且源代码不开放。在这种情况下, 选择了 RTLinux 操作系统, 它是基于标准 Linux 的具有硬实时特性的实时操作系统, 自诞生以来已被广泛应用于基于计算机或者嵌入式系统的实时控制中<sup>[6]</sup>。

该 AFM 控制程序与 RTLinux 的系统结构如图 1 所示。

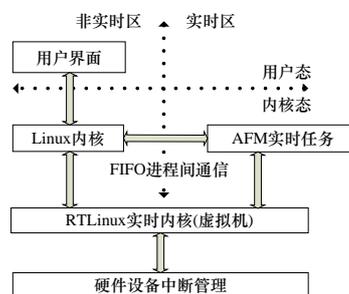


图 1 AFM 控制程序与 RTLinux 系统结构

**基金项目:** 国家自然科学基金资助项目(60574027); 新世纪优秀人才支持计划基金资助项目(NCET-06-0210)

**作者简介:** 周娴玮(1982—), 男, 博士研究生, 主研方向: 纳米定位、控制、操作; 方勇纯, 教授、博士生导师; 董晓坤、张玉东, 硕士研究生

**收稿日期:** 2007-09-16 **E-mail:** zhouxw@robot.nankai.edu.cn

本文设计的 AFM 控制程序与 RTLinux 的系统结构中, RTLinux 实现了一个高效的可抢先实时调度器, 它把 Linux 内核当作此调度器一个优先级最低的进程, 且可被实时进程抢占。当 AFM 实时任务需要运行时, RTLinux 可以抢夺非实时任务, 运行实时任务; 没有实时任务时, RTLinux 就运行 Linux 的非实时任务(界面更新等)。

### 3 控制系统硬件结构

在 AFM 中, 控制系统完成的主要任务概括如下: 采集 AFM 的光斑电压  $z$ , 并使用设计好的控制算法, 计算压电陶瓷的输出电压, 使得光斑电压稳定在设定值。考虑到速度与精度方面的要求, A/D 卡采用 Adlink 公司的 PCI-9111HR, 它具有 16 个通道, 转换时间为  $8\mu\text{s}$ ; 相应地, D/A 卡采用的是 Adlink 公司的生产的 PCI-6208, 它具有 8 个通道。AD/DA 卡的分辨率均为 16 位, 即使对于伸缩范围为  $200\mu\text{m}$  的压电陶瓷管也能达到高于纳秒级的分辨率。在控制过程中, A/D 卡输入为光斑上下电压, 而 D/A 卡的输出信号则为压电陶瓷在 Z 方向的控制电压。由于 RTLinux 的实时特性与硬件密切相关, 本系统选用较高档次的计算机, 其 CPU 是 Pentium 4 2.66 GHz, 内存为 512 MB。修改后的原子力显微镜系统结构如图 2 所示, 灰色的部分为新增加的系统。其中,  $U_{z1}$  表示 AFM 中原有 PI 控制器在 Z 方向上的控制电压, 将控制器的比例和积分参数设置为 0, 从而关掉这个控制信号  $U_{z1}$ (即  $U_{z1}=0$ )。  $U_{z2}$  表示新增控制系统的反馈电压, 它可以通过各种先进的控制方法, 如  $H_\infty$  及自适应算法等来得到, 最终达到提高测量性能的目标。

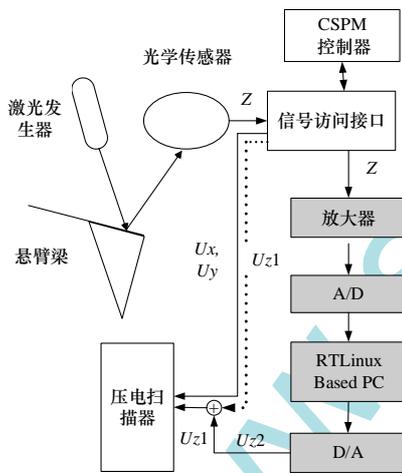


图 2 修改后的原子力显微镜

Adlink 公司提供的驱动程序支持在 Linux 下对 AD/DA 的读写, 但是该驱动程序是一个静态库函数, 不能支持在 RTLinux 下内核态的开发。为此, 必须使用 Linux 内核态下提供的 PCI 函数库自行开发 AD/DA 驱动程序。它主要的步骤是利用 `pci_find_device()` 函数找到指向 AD/DA 资源的指针, 然后利用 `pci_resource_start()` 找到 AD/DA 卡的基地址, 经过初始化后, 就可以使用 `inw()` 和 `outw()` 函数对 AD/DA 的寄存器进行读写操作。

### 4 控制系统软件设计

本系统的控制软件分成 2 部分: 实时部分和非实时部分。实时模块运行在 RTLinux 内核态, 它与硬件(AD/DA 卡)直接进行交互, 它通过 A/D 卡读取光斑的电压值  $z$ , 然后根据设计的控制算法来计算压电陶瓷在 Z 方向的电压  $U_{z2}$ , 并通过

D/A 卡输出。而非实时模块则主要负责界面的数据显示与存储、接收用户的命令(如设置参考点、调节 PID 参数等)并进行响应。实时部分与非实时部分之间的通信通过 RT-FIFO 来实现。

#### 4.1 AFM 控制程序

AFM 控制程序包括数据的采集、数据处理、控制算法和数据的输出, 这是本文编写 AFM 实时反馈系统软件的核心部分。它主要包含 `init_module()`, `cleanup_module()` 和实时采集与处理线程这 3 个函数。当实时任务模块第 1 次加载到内核时, 首先调用的是 `init_module()` 函数, 它完成任务的初始化工作(包括创建实时 FIFO、创建实时线程、设置线程的调度优先级和周期等)。RTLinux 提供的 `pthread_make_periodic_np()` 函数设定实时任务为周期, 且周期的长短可以在参数中设定, 单位为 ns。当实时线程开启后, 它每隔一定时间就会自动执行, 完成光斑电压采集、控制量计算和输出等任务, 然后将 CPU 的使用权交还给非实时任务, 直到下一个周期的来临。`cleanup_module()` 函数负责在系统退出时释放资源(如 AD/DA 卡、内存)。

#### 4.2 界面显示模块

界面显示模块完成数据的预处理、存储、显示以及 PID 算法的参数设置等。在开发界面方面, 采用 QTDesigner 进行。在非实时程序中, 调用 `select()` 函数来检查设备文件 `rtf`(实时 FIFO)是否忙, 以避免读设备文件 `rtf` 时出现阻塞现象。非实时部分的源代码通过 gcc 编译成可执行文件。在加载实时模块后, 再在 Linux 环境下即可直接运行此可执行文件。

#### 4.3 实时和非实时模块的通信

图 3 显示了整个控制系统的流程。

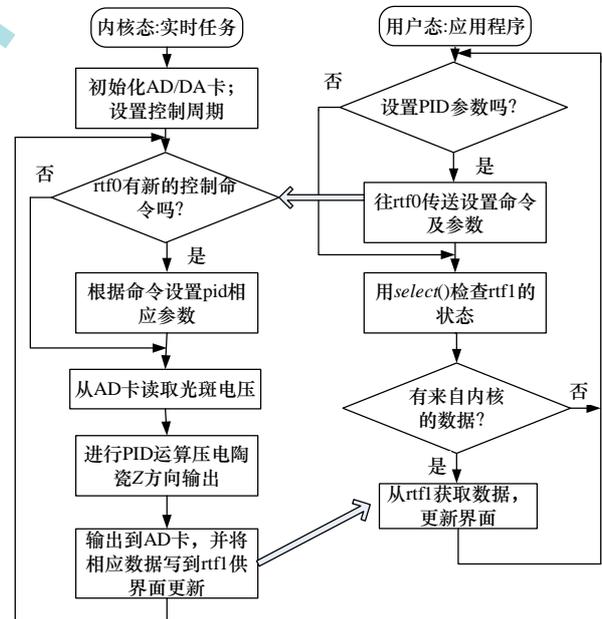


图 3 基于 RTLinux 的 AFM 控制系统流程

采用了 FIFO 来实现 AFM 控制程序和界面显示之间的通信。它的实现主要包括创建、读/写操作、释放 3 个步骤。在 AFM 控制程序中, 主要创建了 2 个实时 FIFO, 其中 `rtf0` 把命令和参数(如 PID 的参数、设定点等)从界面往内核传递; `rtf1` 把数据(如控制输出、光斑电压等)从内核往界面以便非实时任务进行数据的显示或者保存。在 AFM 控制程序中, 通过调用 RTLinux 提供的 API 函数 `rtf_create()` 和 `rtf_destory()`

来创建和撤销实时 FIFO, 并利用 *rtf\_put()* 往 *rtf1* 中写入数据; 为了保证实时线程能及时响应从界面发送来的控制字, 在实时进程中通过函数 *rtf\_create\_hander()* 来建立跟踪 *rtf0* 变化的程序。界面显示模块访问实时 FIFO 时, 可以使用字符设备的读/写函数如 *open*, *write*, *read* 和 *close* 等。

## 5 系统的性能测试

### 5.1 系统实时性测试

如前所述, 控制系统的实时性是决定 AFM 扫描速度的主要因素。因此, 首先测试这种基于 RTLinux 的 AFM 反馈控制系统的实时性。在测试过程中, 使控制系统在正常情况下工作, 然后利用 *clock\_gettime()* 函数来获取周期线程每次的启动时间戳(单位为 ns), 相邻周期时间戳的差值就是控制周期的实际时间; 而它与设定的控制周期之间的差值就定义为延时。在实验过程中, 将控制周期分别设置为 40  $\mu\text{s}$ , 50  $\mu\text{s}$ , 80  $\mu\text{s}$ , 100  $\mu\text{s}$ , 200  $\mu\text{s}$ , 计算并分析了连续 10 万个周期的延迟情况, 其结果如表 1 所示。从这些结果可以看出, 当将控制周期设定在 40  $\mu\text{s}$ ~200  $\mu\text{s}$  之内时, 延迟时间一般不超过 1  $\mu\text{s}$ , 而最大延迟均约为 5  $\mu\text{s}$ 。因此, 本系统具有较好的实时性, 并且可以非常准确地实现设定的控制周期。

表 1 10 万个控制周期延时的分布

| 周期/ $\mu\text{s}$ | 无延时/个  | 0 $\mu\text{s}$ ~1 $\mu\text{s}$ /个 | 1 $\mu\text{s}$ ~2 $\mu\text{s}$ /个 | 2 $\mu\text{s}$ ~5 $\mu\text{s}$ /个 |
|-------------------|--------|-------------------------------------|-------------------------------------|-------------------------------------|
| 40                | 39 114 | 57 059                              | 3 527                               | 210                                 |
| 50                | 36 403 | 61 512                              | 2 048                               | 37                                  |
| 80                | 53 924 | 44 022                              | 2 019                               | 25                                  |
| 100               | 65 184 | 33 368                              | 1 428                               | 20                                  |
| 200               | 40 764 | 55 824                              | 34                                  | 12                                  |

### 5.2 反馈控制下的系统阶跃响应

进一步, 通过阶跃响应测试系统的动态特性。为此, 控制周期选择为 50  $\mu\text{s}$ , 而控制算法则采用常见的比例-积分-微分(PID)反馈控制, 图 4 显示了整个系统在闭环下的阶跃响应。从响应曲线可以看出, 系统的上升时间约为 500  $\mu\text{s}$ , 调节时间约为 1 ms, 超调量为 2%, 满足对样品进行低速扫描的要求。这和一般商业的 AFM 没有很大的差别。但是, 当使用高速扫描时, 如果样品表面起伏较大, 则这种 PID 反馈控制方法无法使系统快速稳定, 此时就需要设计更为先进的控制器来缩短系统的响应时间。

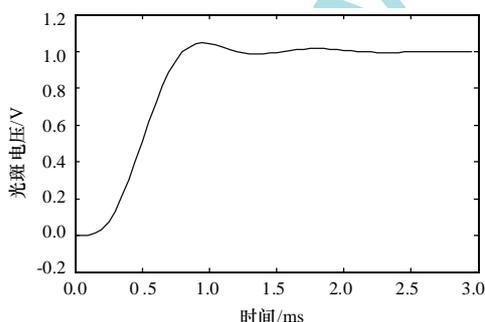


图 4 实时控制下的 AFM 阶跃响应

### 5.3 利用反馈系统成像

为了进一步测试系统的性能, 对标定光栅进行了扫描。

实验所采用的标定光栅(MicroMasch Inc., USA)台阶高度为  $100 \pm 10$  nm, 宽度为 280 nm。图 5 显示了利用本系统在 1 Hz 扫描频率下对光栅的扫描成像, 从图中可以看出扫描图像与样品表面的形貌非常吻合。利用厂家提供的图像后处理软件进行高度分析, 其台阶高度约为 104 nm, 在可接受的误差范围内, 表明系统具有良好的测量性能。

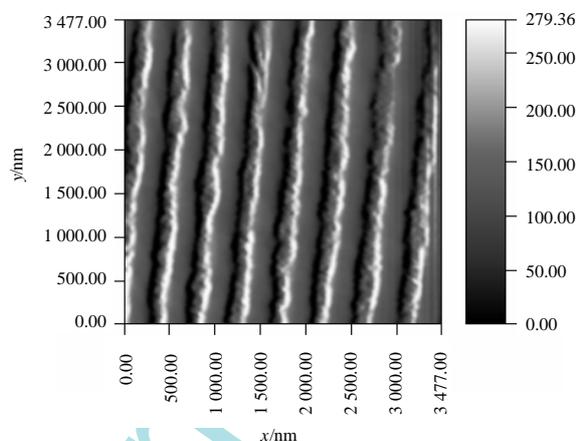


图 5 标定光栅 AFM 图象

## 6 结束语

针对原子力显微镜系统, 本文设计并实现了一种基于 RTLinux 的 AFM 实时反馈控制系统。主要包括系统总体方案设计、硬件结构选择、控制软件开发等 3 个部分。测试结果表明, 这种基于 RTLinux 的 AFM 实时反馈系统, 控制周期延时最大不超过 5  $\mu\text{s}$ , 一般不超过 1  $\mu\text{s}$ , 因此具有较好的实时性。此外, 利用本文所设计的系统对标定光栅进行了扫描成像, 结果表明系统具有良好的测量性能。在后续工作中, 将利用自适应、 $H_\infty$  等先进控制算法来提升平台的测量性能。除此之外, 还将进一步提高平台的开放性, 为进行纳米操作方面的研究提供条件。

### 参考文献

- [1] 白春礼. 扫描隧道显微术及其应用[M]. 上海: 上海科技出版社, 1991.
- [2] Sarid D. Scanning Force Microscopy[M]. New York, USA: Oxford University Press, 1994.
- [3] Zou Qingze, Leang K K, Sadoun E, et al. Control Issues in High-speed AFM for Biological Applications: Collagen Imaging Example[J]. Asian Journal of Control, 2004, 6(2): 164-178.
- [4] Schitter G, Allgower F, Stemmer A. A New Control Strategy for High-speed Atomic Force Microscopy[J]. Nanotechnology, 2004, 15(1): 108-114.
- [5] Wu Ying, Zou Qingze. Iterative Control Approach to Compensate for the Hysteresis and the Vibrational Dynamics Effects of Piezo Actuators[C]//Proc. of the 2006 American Control Conference. Minneapolis, Minnesota, USA: [s. n.], 2006: 424-429.
- [6] Barabanov M. A Linux Based Real-time Operating System[D]. [S. l.]: New Mexico Institute of Mining and Technology, 1997.