技术

免费发布

电工吧 > 资讯 > 市场动态

浅谈PID微分器与滤波器

作者: 电工吧 日期: 2020-10-14 09:19:49 人气: 244 栏目: 市场动态

评论

分享

★ 智能酒店客控系统方案 ★ 办公室照明解决方案 ★ 学校照明解决方案 ★ 展厅照明解决方案 ★ 路灯解决方案 ★ 小区景观解决方案 🤿

0 前面的话

这篇文章肝了好久,控制有时候给人的感觉是披着数学外衣的,但是脱下外衣,发现还是控制,本文有一些基本 的推导,无非是为最后的算法C实现做铺垫,最终目的是能在实践中进行应用和系统调优。

目录

0 前面的话

目录

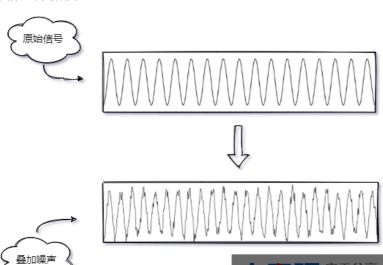
- 1 先说噪声
- 2 噪声对于系统的影响
- 3 对于PID控制器的影响
- 4加入滤波器
- 4.1 传递函数
- 4.2 串联微分的等效形式反馈积分
- 5 C语言实现
- 6 参考
- 1 先说噪声

在电子设备等电路系统中,噪声是不被系统需要的电信号;电子设备产生的噪声会由于多种不同的影响而产生很 大的差异。在通信系统中,噪声是一个错误或不希望出现的随机干扰从而作用于有效的信号。

2 噪声对于系统的影响

噪声出现的第一个场景,当我们在教室里做英语听力,然后旁边的同学手机忽然来了一条短信,这时候往往可以 听到放英语听力的喇叭会被干扰,然后会发出哔哔哔的声音;

下面是一个正弦信号跌加噪声的例子,在原始信号上叠加一定幅度的高斯噪声,可以看到信号不再像原来的正弦 信号那样完美,具体如下图所示;



行业资讯 市场动态 行业研究

在这里搜索..

搜索

热门标签:

今日推荐

2020中国电动车质量排名前十名有哪些品 牌?



2020随着电动车 越来越受年轻上班 族的喜爱, 而骑电 动车也成为了一种 时尚, 电动车作为

毕业5年大学生浅谈自己到电工之路——降龙十... 传Silicon Labs尝试分拆并出售模拟芯片业务 变压器故障及其保护功能基本要求 苹果建议用户恢复 iPhone 和 Apple Watch...

独家专栏



音质好的无线蓝牙耳机 学生党适用

隐私是Android Q的重点关注领域 电工作业不慎手被电弧烧伤的真实故事 特斯拉又开始降价了,部分车型被移除包括... arduino控制继电器 不使用IOS企业签名的将会很难 交流接触器常用接线电路图和实物图 业界最高性能移动设备5G WiFi解决方案 细分引风机启动注意事项及停运步骤

精彩推荐



激光烟雾净化器哪 家好?



一级二级和三级配 电箱的区别和技术



合表用电是什么情 况

热门排行

本原理

自动低频减载的基

家装明线怎么铺线才能又快又好

微机测试系统的信号

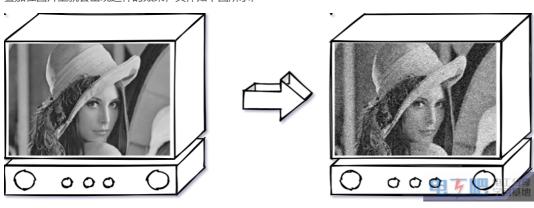
爆闪灯是干什么用的_爆闪灯的结构

盐穴再利用建压缩空气蓄能电站 防火防盗防老王? 这些智能家居安防工具...

被遗忘的发明:史上第一辆配备无线设备...



或者,很久很久以前,数字电视还没有普及,那时候的显像管的黑白电视,也容易出现这样的雪花一样的噪声,叠加在图片上就会出现这样的效果,具体如下图所示;



从上述的例子中可以看到,噪声往往会对系统造成一定程度的影响,但是如果噪声的幅度减小到一定程度,对于 系统的影响可能就没有那么容易被发现。

下面做一个实验;在一张黑色图片上叠加幅度很小幅度的高斯噪声;从第二张图片中发现噪声没有影响到整体图片;

然后我尝试提高了整幅图片的亮度,发现,噪点便开始出来了,这像极平时那些枪版影片的马赛克画质;整体的 实验结果如下图所示;



3 对于PID控制器的影响

既然噪声的幅度减小到一定程度,对于系统的影响可能就没有那么容易被发现,那么对理想的PID控制器又有什么影响呢?

不要忘了,在理想PID控制器中,微分控制器会对偏差的变化率(斜率)进行累加,从而产生积分器的输出;

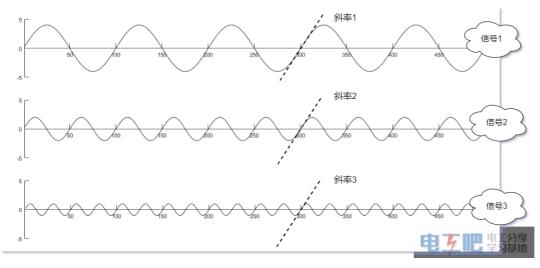
对于微分器来说,即使噪声幅度足够小,但是只要达到足够高频率,偏差的变化率一样可以变得很大,下面举个 例子;

这里有一个固定频率和赋值的噪声为信号1,这个信号可以表示为:

于是我们尝试将信号1的幅度减半,频率变为原来的两倍,得到了信号2:

以此类推,在信号2的基础上,幅度再减半,频率乘以2,得到信号3:

简单画了一下这个信号,具体如下图所示;



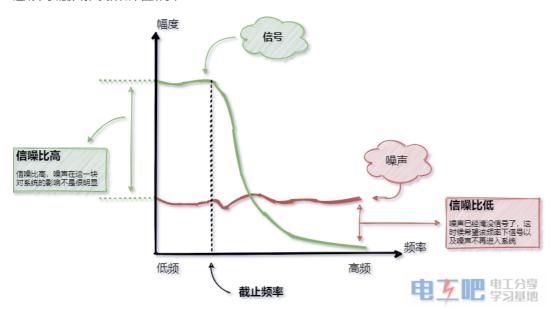
假设分别取三个信号位于该点的斜率,从图中可以看到,斜率1,斜率2,斜率3是相同的,简单验证一下,在时刻,可以得到:

所以这里就是求复合函数的微分,由于选取的点比较特殊,发现最终计算得到的结果,因此也可以发现,即使减小了噪声的幅度,但是对于较高频率的噪声,依然会产生较大斜率。

国产光刻机与荷兰ASML光刻机的具体差距... 【高性能|操作简便】实现1/100mm的高... 遇到高频噪声,那么微分器会产生较大的输出,从而最终对系统造成影响,这是我们不希望出现的结果,因此在 反馈回路中并不希望高频噪声进入PID控制器的计算,这里就需要低通滤波器将噪声滤除。

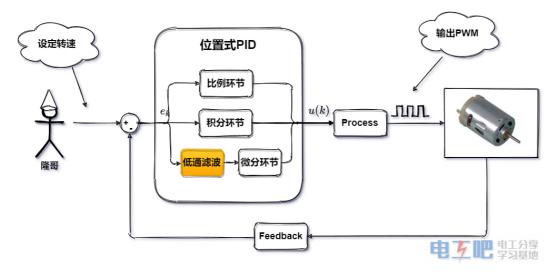
4 加入滤波器

低通滤波器可以滤除高频信号,这样保留了有效信号,可以设置所需的截止频率;系统处理有效信号,由于低频部分信噪比较高,因此噪声对于系统的影响较小,而高频部分,信噪比就很低,这时候对于系统来说,噪声就会造成不小的影响,具体如下图所示;



信噪比:有效信号和噪声的比值,英文名称叫做SNR或S/N (SIGNAL-NOISE RATIO);

所以下面我们会在PID控制器的微分部分加入低通滤波器,这样对反馈的信号进行一部分处理,从而减小系统干扰,如下图所示;



4.1 传递函数

概念: 拉普拉斯变换是对于 函数值不为零的连续时间函数 通过关系式 (式中为自然对数底的指数)变换为复变量的函数。它也是时间函数的"复频域"表示方式。

也就是说拉式变换可以将时域关系变换到频域中,这样可以便于系统进行分析。

下面是本文下面会用到的时域函数对应的拉普拉斯变换:

积分:

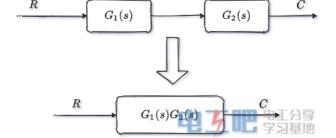
微分:

低通滤波器的传递函数:

低通滤波器中的截止频率即为,单位是;

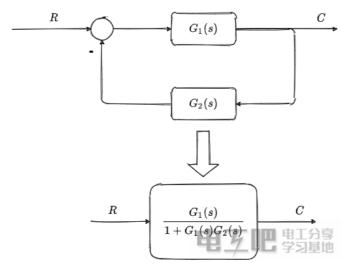
4.2 串联微分的等效形式反馈积分

串联等效传递函数的关系为,两个方框串联等于各个方框传递函数的乘积;具体如下所示;

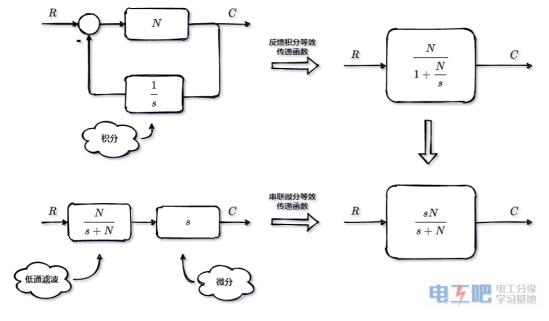


因此低通滤波串联微分的传递函数为:

闭环负反馈的等效传递函数的关系如下所示;



这里我们可以使用负反馈积分的方式,构建等效于串联微分的传递函数,最终的传递函数结果是相同的,具体如下图所示;

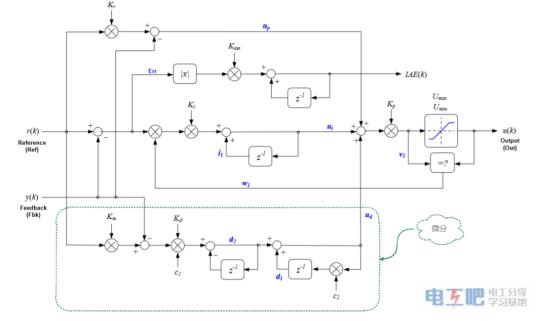


串联微分的形式,可能在算法的实现上会更加直观,但是会比较费资源;

使用负反馈积分的等效形式进行实现,则进一步减少了算法的资源消耗,下面给出一个TI公司的PID算法实现就是通过负反馈积分的等效形式进行实现的。

5 C语言实现

这里直接使用了TI公司的PID算法,对于微分部分做了滤波的处理,并且使用的是负反馈积分的方式,具体可以参考controlSUI<u>TE</u>libsapp_libsmotor_controlmath_blocksv4.2pid_grando.hPID控制器的整体框图如下所示,我们只关心微分部分;



首先可以发现满足:

这里滤波器有两个系数和,它们必须满足截止频率 (单位Hz) 和采样周期 (单位秒) 以下的关系:

C语言实现如下:

```
========= File name: PID GRANDO.H ===========================
========*/ #ifndef PID
H #define PID H typedef struct { iq Ref; // Input: reference set-point iq Fbk; // Inpu
            iq Out;
                      // Output: controller output
                                                 iq c1;
                                                         // Internal: derivative filter coeffi
        iq c2; // Internal: derivative filter coefficient 2 } PID TERMINALS; // note: c1 & c2 pl
aced here to keep structure size under 8 words typedef struct { _iq Kr; // Parameter: reference set-poi
                                                                // Parameter: integral g
              _iq Kp; // Parameter: proportional loop <u>gai</u>n _iq Ki;
nt weighting
ain ig Kd;
                 // Parameter: derivative gain
                                            _iq Km;
                                                       // Parameter: derivative weighting
   _iq Umax; // Parameter: upper saturation limit __iq Umin; // Parameter: lower saturation limit
 } PID_PARAMETE<u>RS</u>; typedef struct { _iq up; // Data: proportional term
                                                                 _iq ui; // Data: integr
         _iq_ud; // Data: derivative term _iq_v1; // Data: pre-saturated controller output _i
al term
q i1; // Data: integrator storage: ui(k-1) iq d1; // Data: differentiator storage: ud(k-1)
2; // Data: differentiator storage: d2(k-1) _iq w1; // Data: saturation record: [u(k-1) - v(k-1)]
 } PID_DATA; typedef struct { PID_TERMINALS term; PID_PARAMETERS param;
                                                                       PID DATA dat
a; } PID CONTROLLER; /*------ Defa
ult initalisation values for the PID objects -----
                   #define PID TERM DEFAULTS {
                                0,
                                         0
                                                    } #define PID PARAM DEFAULT
                                                              _IQ(0.0),
S {
                  IQ(1.0),
                                        IQ(1.0),
              IQ(0.0),
                                                          IQ(1.0),
                                                                                IQ(-
              } #define PID DATA DEFAULTS {
1.0)
                                                           IQ(0.0),
                                                                                 IQ
                   IQ(0.0),
(0.0),
                                         IQ(0.0),
                                                               IQ(0.0),
              _IQ(0.0),
                                    IQ(0.0),
                                                          IQ(1.0)
                                   ----- PID Macro Definition ------
     -----*/ #define PID MACRO(v)
portional term */ v.data.up = IQmpy(v.param.Kr, v.term.Ref) - v.term.Fbk;
                                                                          /* integral t
           v.data.ui = IQmpy(v.param.Ki, IQmpy(v.data.w1, (v.term.Ref - v.term.Fbk))) + v.data.i
                                  /* derivative term */
                                                        v.data.d2 = _IQmpy(v.param.Kd, _
1; v.data.i1 = v.data.ui;
IQmpy(v.term.c1, (IQmpy(v.term.Ref, v.param.Km) - v.term.Fbk))) - v.data.d2; v.data.ud = v.data.d
              v.data.d1 = IQmpy(v.data.ud, v.term.c2); /* control output */
ata.v1 = IQmpy(v.param.Kp,
                            (v.data.up + v.data.ui + v.data.ud));
                                                          v.term.Out= IQsat(v.data.v
1, v.param.Umax, v.param.Umin); v.data.w1 = (v.term.Out == v.data.v1) ? _{IQ}(1.0) : _{IQ}(0.0); #endif // _{IQ}(0.0)
PID H
```

6 参考

https://en.wikipedia.org/wiki/Low-pass_filter

自动控制原理 第五版 胡寿松 P47

虽然写的不一定是最好,但是每一个字、每一个公式都是用心码的,每一张图都是用心画的,每一句话都是加入 了自己的理解;另外笔者能力有限,文中难免存在错误和纰漏,望轻拍指正。

责任编辑:xj

原文标题: PID微分器与滤波器的爱恨情仇

文章出处: 【微信公众号: 传感器技术】欢迎添加关注! 文章转载请注明出处。

本文标签:

5

以上电工吧小编收集整理的 浅谈PID微分器与滤波器 部分内容来自网络,如有侵权请联系删除

本文地址: http://www.dg8.com.cn/news/9192.html

end

上一篇 2020年将成为世界半导体交易金额第二高的年份

下一篇 在遥远的"人机"预言实现前,脑机接口技术应先回到最初的医学使命

相关推荐

Waymo:告别"无人驾驶"暗示是特斯拉所导致的

家里的双控双开开关怎么接比较好

Fluke TiX640红外热像仪的特点特性及功能应用

京东方将于年底向苹果 iPhone 12 系列供应屏幕;新材料让锂 Linux内核的国产系统

2017年智能家居创业, 你必须了解的一些事

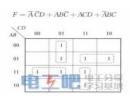
空调外机常年暴露在户外要不要盖住

"不造车"战略成新宠, 美的 vs 华为, 两者是否类似?

关于叶面积分析系统(智能叶面积仪)产品的相关介绍

TCL新风空调C挂机荣获2020中国家电"好产品"奖:实至名

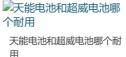
热门推荐



卡诺图化简画圈的原则和步



注意! 电工证查询系统的入 口地址变更了

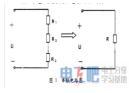




无线路由器耗电量大吗?路 由器一个月会耗电多少



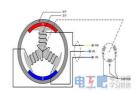
2020中国电动车质量排名 前十名有哪些品牌?



等效电阻的三种求法



电烙铁内热外热哪个好?电 烙铁内热和外热的区别



无刷直流电机【BLDC】与 DD直驱变频电机哪个好?