

# D轴暂态开路时间常数辨识及应用研究

汇报人：

2024-01-18

| CATALOGUE |

# 目录

- 引言
- D轴暂态开路时间常数辨识原理
- D轴暂态开路时间常数辨识算法设计
- D轴暂态开路时间常数辨识实验验证
- D轴暂态开路时间常数在电力系统中的应用
- 结论与展望

# 01 引言





# 研究背景和意义

## 1

### 电力系统稳定性

D轴暂态开路时间常数是电力系统稳定性分析的重要参数，对于保障电网安全稳定运行具有重要意义。

## 2

### 新能源并网

随着新能源的大规模并网，电力系统的动态特性变得更加复杂，对D轴暂态开路时间常数的准确辨识提出了更高的要求。

## 3

### 控制器设计

在电力系统稳定器（PSS）等控制器的设计中，需要准确获取D轴暂态开路时间常数以提高控制性能。





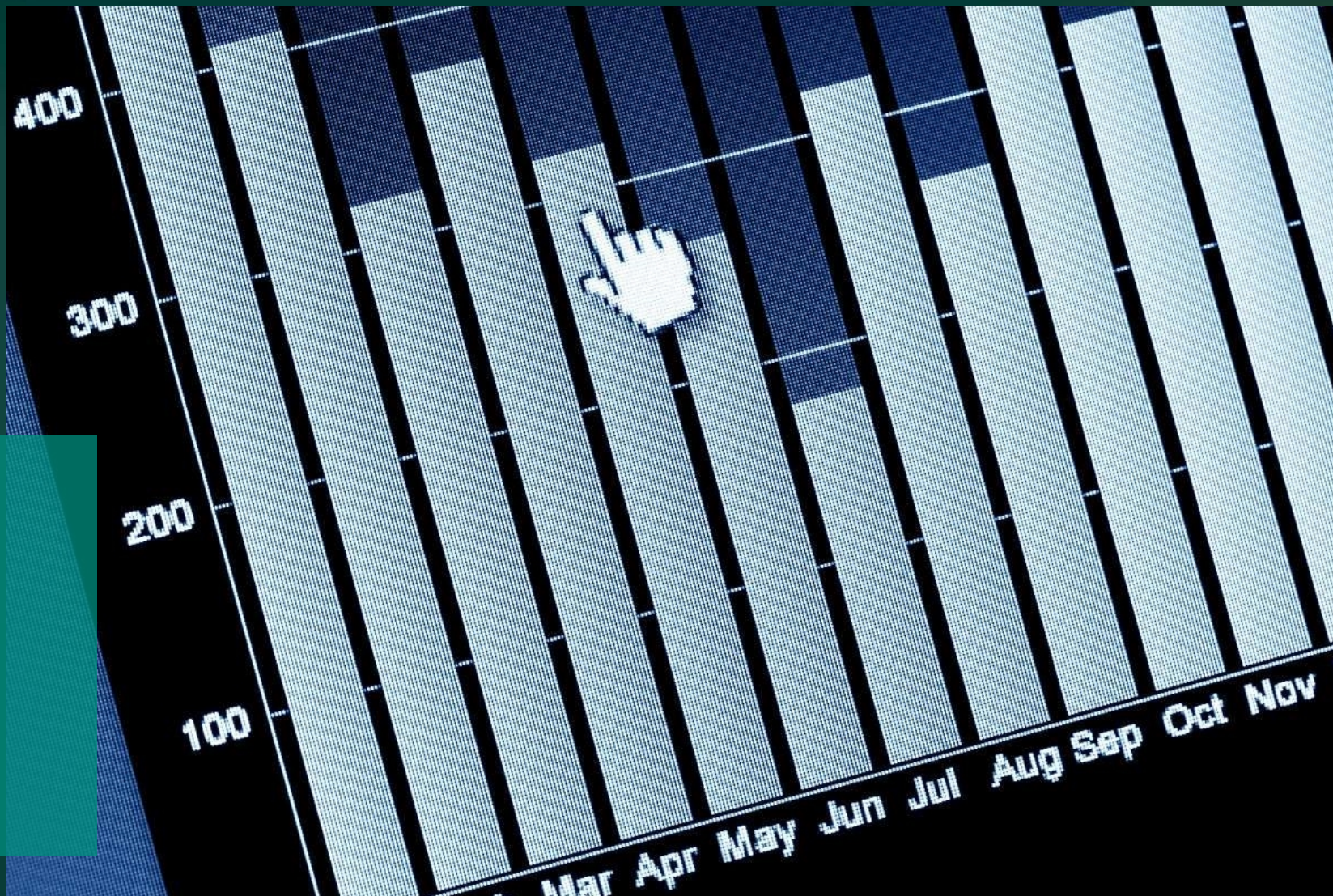
# 国内外研究现状及发展趋势

## 研究现状

目前，国内外学者已经提出了多种D轴暂态开路时间常数的辨识方法，包括时域仿真、频域分析和智能算法等。

## 发展趋势

随着计算机技术和人工智能的发展，基于数据驱动的D轴暂态开路时间常数辨识方法将成为未来研究的热点。





# 本文主要研究内容和创新点



## 研究内容

本文旨在提出一种基于深度学习的D轴暂态开路时间常数辨识方法，并通过仿真和实验验证所提方法的有效性和优越性。

## 创新点

本文的创新点在于将深度学习技术应用于D轴暂态开路时间常数的辨识中，克服了传统方法在处理非线性、时变系统时的局限性，提高了辨识的准确性和效率。

02

# D轴暂态开路时间常数辨识原理





# D轴暂态开路时间常数定义及物理意义

## D轴暂态开路时间常数定义

D轴暂态开路时间常数是指在电力系统中，当D轴发生暂态开路时，系统从初始状态到达稳定状态所需要的时间。它是描述系统暂态过程的重要参数。

## 物理意义

D轴暂态开路时间常数反映了系统在D轴暂态开路下的响应速度和稳定性。较小的时间常数意味着系统能够更快地达到稳定状态，而较大的时间常数则表明系统需要更长的时间来恢复稳定。





# 辨识原理及方法概述

## 辨识原理

D轴暂态开路时间常数的辨识原理基于系统的动态响应特性。当D轴发生暂态开路时，系统的动态响应会发生变化，通过测量和分析这些变化，可以辨识出D轴暂态开路时间常数。

## 方法概述

辨识D轴暂态开路时间常数的方法主要包括时域分析法、频域分析法和基于智能算法的方法。时域分析法通过直接观察系统动态响应的时域波形来辨识时间常数；频域分析法利用系统的频率响应特性进行辨识；基于智能算法的方法则运用优化算法、神经网络等技术进行辨识。



# 基于不同方法的辨识效果比较

## 时域分析法

时域分析法具有直观、简单的优点，能够直接观察系统动态响应的变化。然而，该方法对测量设备的精度和采样率要求较高，且容易受到噪声干扰的影响。

## 频域分析法

频域分析法能够准确地提取系统的频率响应特性，对于线性时不变系统具有较好的辨识效果。但是，该方法需要较复杂的数学计算和信号处理技术，且对于非线性系统的辨识效果有限。

## 基于智能算法的方法

基于智能算法的方法具有较强的自适应能力和鲁棒性，能够处理复杂的非线性系统和噪声干扰问题。然而，该方法需要大量的训练数据和计算资源，且对于不同系统的通用性有待提高。

03

# D轴暂态开路时间常数辨识算法设计





# 算法整体架构设计



```
while (p < sizeof(unsigned int) && count > 0) {
    if (put_user(*(char *)(&sample_step + p), buf))
        return -EFAULT;
    buf += p * 4; count--; read += 4;
}

pnt = (char *)prof_buffer + p - sizeof(atomic_t);
if (copy_to_user(buf, (void *)pnt, count))
    return -EFAULT;
read += count;
*opos += read;
return read;
}

/* Writing to /proc/profile resets the counters
 * Writing a profiling multiplier value into it also sets the profiling
 * interrupt frequency, on architectures that support this
 */
static ssize_t write_profile(struct file *file, const char __user *buf,
                             size_t count, loff_t *opos)
{
    #ifdef CONFIG_SMP
    extern int setup_profiling_timer(unsigned int multiplier);
    #endif

    if (count == sizeof(int)) {
        unsigned int multiplier;

        if (copy_from_user(&multiplier, buf, sizeof(int)))
            return -EFAULT;

        if (setup_profiling_timer(multiplier))
            return -EINVAL;
    }
}

return count;
}

profile_discard_fop_buffers();
memset(prof_buffer, 0, prof_len * sizeof(atomic_t));
return count;
}
```



## 01

### 输入输出设计

确定算法的输入和输出，包括原始信号、处理后的信号、辨识结果等。

## 02

### 模块划分

将算法划分为数据预处理、特征提取、模型训练、参数优化等模块。

## 03

### 流程设计

设计算法的整体流程，包括模块间的调用关系和数据传递方式。



# 数据预处理及特征提取方法

## ● 数据预处理

对原始信号进行去噪、滤波、归一化等处理，提高信号质量。

## ● 特征提取

提取与D轴暂态开路时间常数相关的特征，如电压、电流、功率等的变化率和波形特征。

## ● 特征选择

从提取的特征中选择对辨识结果影响最大的特征，降低模型复杂度。



以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：  
<https://d.book118.com/178053003044006076>