

“飞思卡尔”杯全国大学生智能车邀请赛仿真平台

Plastid2 智能车仿真平台

三角洲工作室 2007.3



***DELTA
STUDIO***

前言

Plastid2 是为“飞思卡尔”杯全国大学生智能车邀请赛开发的新一代智能车仿真系统，在上一版本仿真系统基础之上，不仅可以针对不同的赛道、赛车、路径识别方案、控制策略等内容进行仿真和相关分析，还增添了许多新的功能，使仿真系统更趋近于实际情况，为用户提供更好、更真实的虚拟仿真平台。

Plastid2 继承并拓展了上一版本的功能特色，主要有以下几大特点：

1、赛道与赛车环境模拟

系统分别针对赛道与赛车建立模型，使用者可以方便地自行设计直线、弯道等各种形状的赛道，并可根据赛车的实际情况调整赛车的实际参数，使用灵活方便。在条件限制，没有办法制作实际试验赛道或智能车尚未制作完成的情况下，更可以在该系统中验证调试控制算法。

2、控制算法仿真验证

系统采用纯软件仿真方式，通过将控制程序编写成 `dll`，系统调用 `dll` 进行仿真。`dll` 的编写可以使用 VC6、VC2005、Delphi7、Delphi2006，使用者可以根据自身情况，选择最适合自己的编程环境编写控制程序。验证调试后的算法代码，也可以很方便的移植到单片机程序中。

3、路径识别方案分析

系统提供了广泛使用的光电传感器和 CCD 传感器模型，使用者可以自行设计传感器的数量和排列方式、位置，在系统中进行仿真，经过分析比较，从而获得优化方案。很大程度上解决了实地试验中更换传感器排布麻烦、耗时的问题，从而极大提高方案分析效率。

本文档旨在对即将使用 Plastid2 的用户给予最简明的帮助，使您能轻松快速的掌握本软件系统的使用。全文将分为赛道、赛车、仿真以及回放四个部分对 Plastid2 地使用进行说明讲解。

由于时间仓促及水平有限，在软件运行和使用中难免会出现一些程序错误，希望您谅解。同时我们也热忱希望您在在使用过程中将遇到的问题和您宝贵的意见及时地反馈给我们。

技术支持及问题反馈：Delta.Plastid@gmail.com

我们将对此致以万分的谢意！

最后，感谢您选择和使用本仿真系统，我们三角洲工作组向您对本软件的支持致以最崇高的敬意！

目录

第一章 软件概况.....	1 -
1.1 运行环境.....	1 -
1.2 版本改进.....	2 -
1.3 主界面概述.....	3 -
第二章 软件架构.....	4 -
第三章 赛道设计.....	5 -
3.1 功能介绍.....	5 -
3.2 设计赛道.....	6 -
3.2.1 赛道文件操作.....	6 -
3.2.2 赛道设计.....	8 -
第四章 赛车设计.....	14 -
4.1 功能介绍.....	14 -
4.2 设计赛车.....	15 -
4.2.1 赛车文件操作.....	15 -
4.2.2 赛车设计.....	16 -
4.3 测试赛车.....	22 -
4.3.1 功能介绍.....	22 -
4.3.2 测试操作.....	23 -
第五章 仿真模拟.....	24 -
5.1 功能介绍.....	24 -
5.2 仿真模拟.....	25 -
5.2.1 加载.....	25 -
5.2.2 仿真模拟.....	26 -
第六章 结果回放.....	28 -
6.1 功能介绍.....	28 -
6.2 结果回放.....	29 -
6.2.1 加载仿真结果文件.....	29 -
6.2.2 回放控制.....	30 -
第七章 默认参数与规则设置.....	31 -
7.1 默认参数设置.....	31 -
7.2 规则设置.....	32 -
附录 A 如何编写控制算法文件.....	33 -
附录 B 如何导出仿真数据.....	35 -

第一章 软件概况

Plastid2 是为“飞思卡尔”杯全国大学生智能车邀请赛开发的新一代智能车仿真系统，在上一版本仿真系统基础之上，不仅可以针对不同的赛道、赛车、路径识别方案、控制策略等内容进行仿真和相关分析，还增添了许多新的功能，使仿真系统更趋近于实际情况，为用户提供更好、更真实的虚拟仿真平台。

作为 Plastid2 软件的简单介绍，本章的主要内容如下：

- 软件系统的运行环境要求
- Plastid2 版本的改进以及新版本特点
- Plastid2 主界面的概述

1.1 运行环境

Plastid2 软件是免安装软件，避免了繁琐的安装过程。并且程序不需要很高的系统配置，使用更加方便。

- 操作系统：Windows 2000 / XP 及以上
- CPU：1.3GHz 及以上
- 内存：
- 光驱：
- 显示分辨率：1024*768 及以上
- 其他：

DELTA
STUDIO

1.2 版本改进

Plastid2 在上一代版本 Plastid 的基础上作了很多的改进,一方面继承了上一版本的分类方法和大多数的仿真方法,同时也在其基础上改进了仿真模型,增加了更多的仿真环境条件,使得程序的实用性更强、仿真效果更好;另一方面,Plastid2 大幅度优化了用户 UI,界面更友好、操作更人性化,克服了上一版本 Plastid 界面操作中出现的许多误操作,并且简洁的外观让用户使用更加舒适、方便,大幅度改善了使用过程中的用户体验。

以下是 Plastid2 的各种特点以及一些新的特性:

1) Plastid2 是一款免安装软件

Plastid2 软件是免安装的,避免了繁琐的安装过程,使用更加方便。直接双击运行 Plastid2.exe, 就可以开始使用 Plastid2 了。

2) 优化的赛道设计

Plastid2 提供了比上一版本更加详细的赛道参数:道宽、线宽、摩擦系数等。实际参数的加入,使 Plastid2 仿真系统更接近于真实赛道环境。

Plastid2 取消了上一版本自由度极大的手绘模式,因为正常比赛用的赛道应该为各处连续的,而手绘模式很难保证绘出的赛道各处连续。因此,Plastid2 取消了手绘模式,只使用直线、圆弧两种简单元素绘制赛道。

另外,Plastid2 的赛道设计界面十分的人性化,更注重赛道的细节,提供了缩放功能,并且在赛道设计主视窗内提供了鼠标取坐标、鼠标移动当前主视窗等功能,使赛道设计更加的方便合理。

3) 更详尽真实的赛车模型

Plastid2 在上一版本的赛车模型基础上,改进了电机模型,增加了相关参数,使得 Plastid2 的仿真结果能够更加接近实际试验的结果,提高开发效率。

4) 扩展的传感器模型

上一版本中,传感器模型中仅提供了工作在非连续模式^①下的光电传感器模型,仿真对象相对单一;Plastid2 中,扩展了原有的光电传感器模型,增加了连续模式,并且增加了 CCD 传感器模型。

当用户采用光感电路来识别路径时,可以按照自己的想法设定传感器的个数、排列的位置,在系统上反复进行仿真试验,从而可以得出效果最好的传感器排列的优化方案,这样可以很大程度上解决实地试验中更换传感器排布麻烦、耗时的问题;而采用 CCD 图像采集的方式进行路径识别时,使用者可以考虑并选择 CCD 的分辨率、安放位置等,系统提供相应参数的设置。

扩展后的光电传感器模型和 CCD 传感器模型,已经基本上覆盖了大赛中所使用的传感器类型,使得 Plastid2 在大赛中面向的对象范围更广泛,适应性更好。

^① 关于光电传感器的连续与非连续两种工作模式,将在“赛车设计”章节中详细介绍。

1.3 主界面概述

双击运行 Plastid2 之后，即见到程序的主界面：

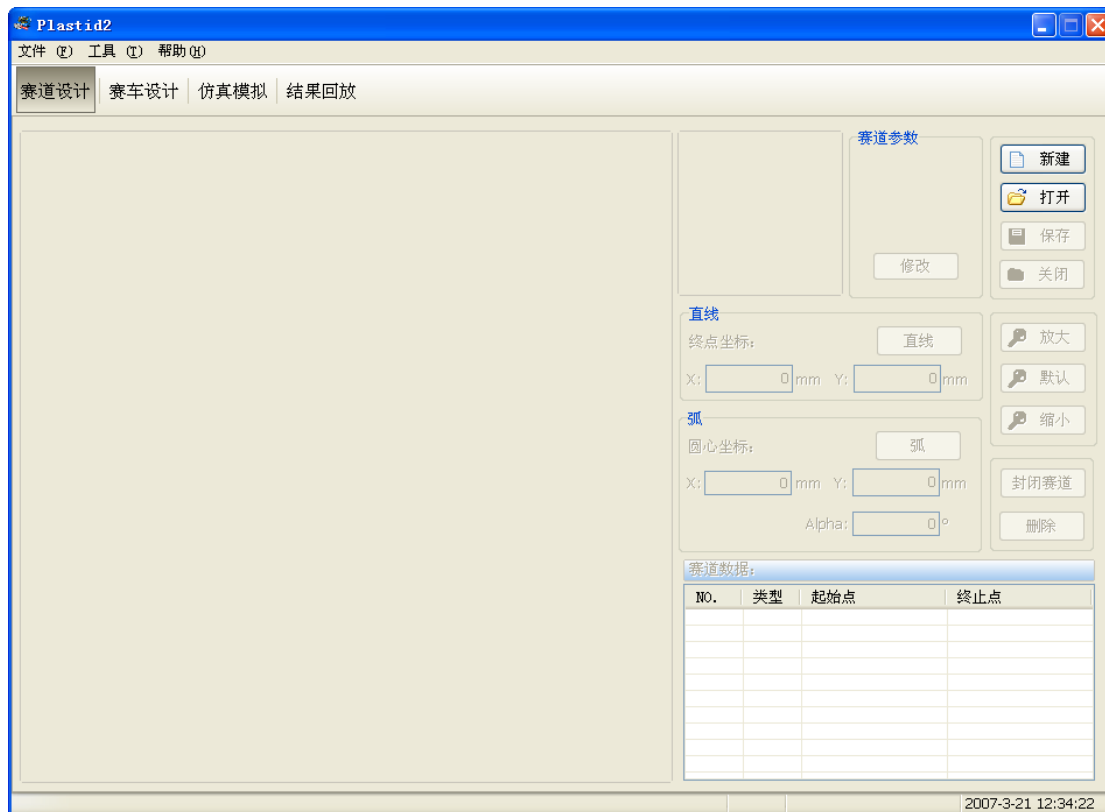


图 1.3.1 程序主界面

在此界面中，用户可以在菜单工具栏中的“文件”、“工具”、“帮助”等菜单进行操作；同时，也可以操作菜单工具栏下方的选项：“赛道设计”、“赛车设计”、“仿真模拟”、“结果回放”，进入相应的操作子界面进行进一步的操作。

赛道设计：

在赛道设计子界面中，可以进行赛道的设计操作，如新建及修改赛道、赛道基本参数设定等。

赛车设计：

在赛车设计子界面中，用户建立自己的小车模型，并根据自己小车的实际情况对相关参数进行设置。同时，我们也提供了一些默认参数供用户调试。

仿真模拟：

完成前面的赛道和赛车设计之后，在仿真模拟子界面中，可以完成仿真系统最重要的功能——仿真模拟，为用户编写的算法作一个定性的评估；

结果回放：

在结果回放子界面中，可以观看仿真模拟之后保存的结果，用户可以通过播放、暂停、步进、步退等功能操作回放过程，分析回放结果。

第二章 软件架构

Plastid2 整个仿真系统的构架图如下图所示，主要分为基本模型层、传感器层、控制算法层、以及仿真环境层。

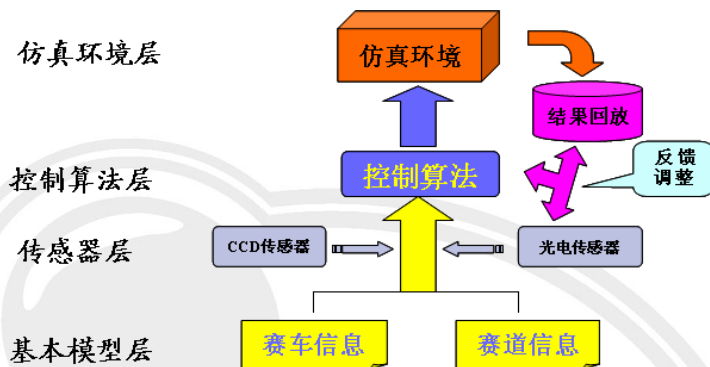


图 2.1 软件框架

基本模型层包括赛车模型与赛道模型，用户可根据实际情况设定模型参数如赛道参数、赛车参数、电机参数等等，它为整个系统提供了底层的物理模型驱动，仿真结果在此基础上计算而得。

传感器层包括传感器种类、数量以及安装位置的设置，用户可以根据自己参赛队的实际的条件选择合适的传感器方案，同时也可以暂时设定一个方案，以后再作进一步的调整。

在控制算法层，用户可以提供自己编写的算法以供仿真使用。

仿真环境层在各个模型参数设置完毕、初始化成功的条件下，通过前面所设置的赛车、赛道模型以及接受控制算法所输出的控制信号（电机控制、转向控制信号），计算出车的行驶路线及各个运动、机电状态参数，并即时地将数据传回控制算法层。

在仿真过程中，系统将仿真过程中的一些重要状态参数记录下来，并可将结果保存为仿真记录文件。在回放模式中，用户可调用仿真保存的仿真记录文件，对其仿真结果进行后期分析和处理，进而改进自己的赛车设置以及控制算法。

第三章 赛道设计

3.1 功能介绍

赛道是整个智能车比赛不可缺少的要素之一，对于一个具有高级控制策略的智能车来讲，应具备高鲁棒性，即在不同的赛道中都保持稳定的发挥。为了验证这一点，在实车试验中，就必须制作不同的赛道进行试验，这样即不经济也缺乏效率，并且实际情况也不完全允许的。然而，Plastid2 提供的赛道设计可以轻松解决这个问题，用户可以在系统中自行设计不同的赛道，然后保存为文件，在仿真时将其调用即可。

在主界面中，點選“赛道设计”，即可进入赛道设计界面（软件进入时的默认界面），如下图所示：



图 3.1.1 赛道设计界面

赛道主视窗及全局视窗可即时显示出正在设计的赛道的形状，供用户设计时参考，当坐标停留在主视窗内时，左下角状态栏会有当前鼠标点的坐标显示，方便用户设计赛道时地定位。

用户可以通过赛道编辑来实现赛道地设计，右下角的数据列表还有当前赛道地数据信息显示，这样使赛道设计更加方便、直观。具体地各项操作将在后面的小节作详细的介绍。

3.2 设计赛道

3.2.1 赛道文件操作

【新建赛道】

点击“新建”按钮，程序自动新建一条空白赛道，新赛道中，除了赛道信息之外^①，其它数据全部为空，用户可以根据需要设计赛道，关于赛道参数的设置将在后面小节介绍。

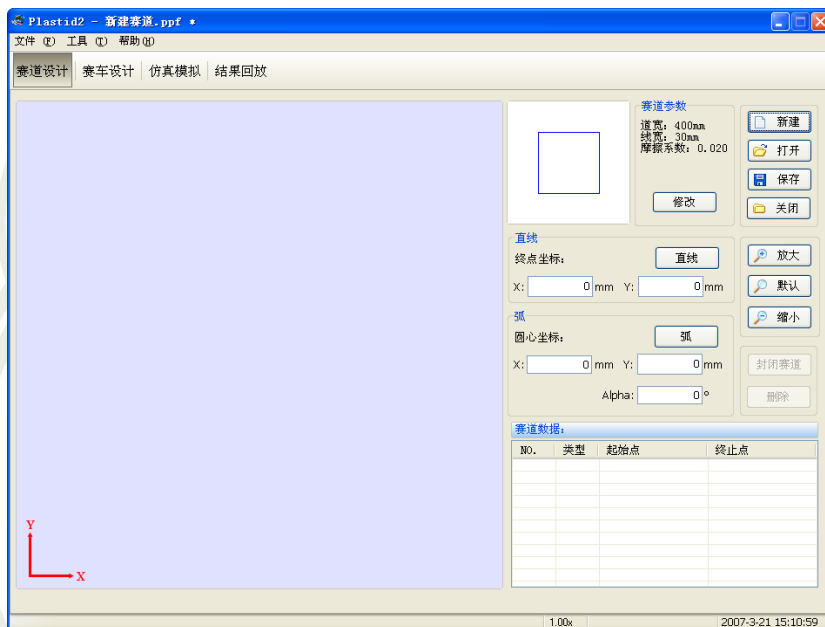


图 3.2.3 新建赛道



提示：赛道主视窗显示的是全局视窗中蓝色方框内的内容。

【打开已有赛道】

Plastid2 中提供有示例赛道，用户刚开始使用时可以打开提供的示例文件。

点击“打开”按钮，弹出浏览窗口，供用户选择需要打开的赛道文件。

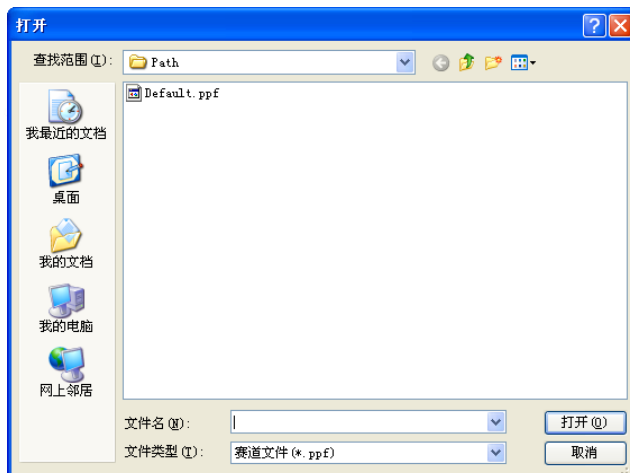


图 3.2.1 打开赛道文件对话框

^① 初始赛道参数来自默认值，其相关设置在“默认参数和规则设置”章节详细介绍。

用户选择已有赛道文件，程序将加载该文件，如下图。

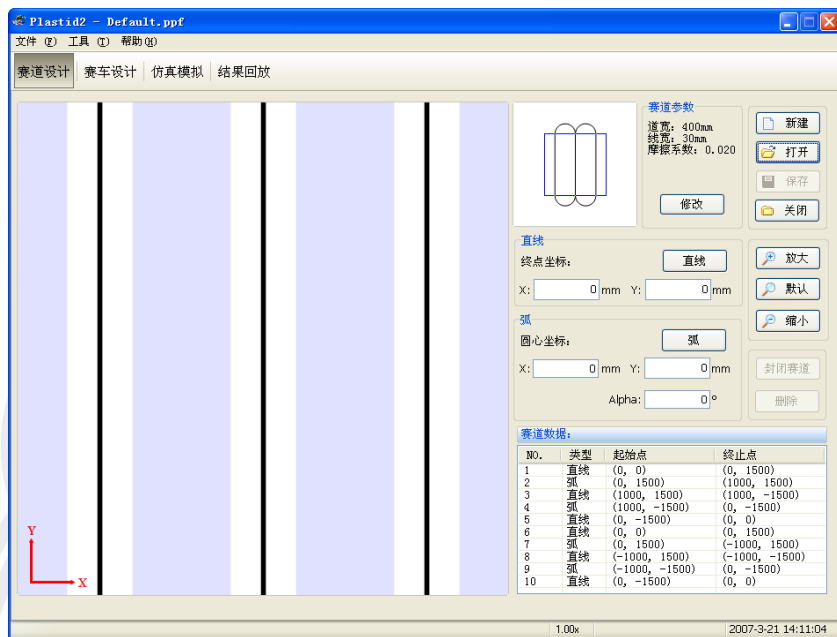


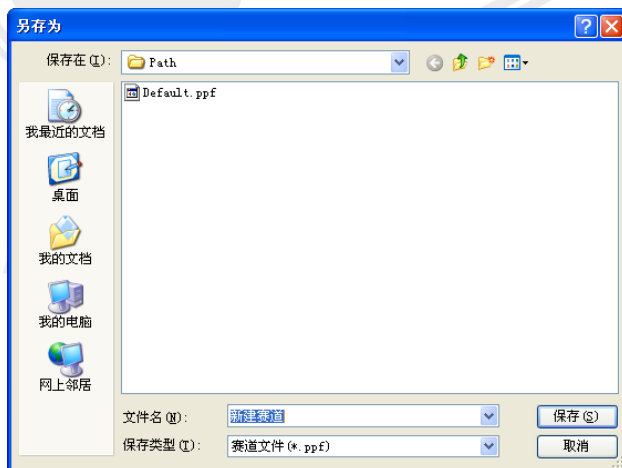
图 3.2.2 打开赛道



提示：程序 Path 文件夹下的 Default.ppf 为程序提供的示例赛道文件，Plastid2 的赛道文件后缀名为 ppf。

【保存赛道】

当用户赛道设计完毕需要对其进行保存时，点击“保存”按钮，在弹出对话框中选定保存位置，输入文件名，点击“保存”按钮，即可完成保存操作。



3.2.4 保存赛道文件对话框

【关闭赛道】

当用户完成当前赛道设计或编辑，不想继续编辑该赛道，希望关闭当前赛道文件时，点击“关闭”按钮。如果当前赛道还有未保存数据，系统将提示用户是否保存当前赛道的数据；如果没有未保存数据，系统将直接关闭当前赛道文件。

3.2.2 赛道设计

对于一个新建的赛道，应该如何进行设计？

首先，了解一下系统关于赛道设计的一些约定。

系统对于赛道位置的描述采用笛卡尔坐标系： X 轴与 Y 轴互相垂直，相对用户向右为 X 轴正方向、向上为 Y 轴正方向；角度为矢量与 X 轴正方向的夹角，逆时针方向为正，顺时针方向为负。

赛道起点坐标为 $(0, 0)$ ，赛道的组成元素为直线和圆弧，赛道各元素之间按添加顺序首尾相连，即用户不能自行指定赛道元素起始点，只能指定终止点。因此，在删除赛道元素时，是将当前选择的元素及之后的所有元素一并删除。

实际中，赛道各元素之间应平滑连续过渡，即直线只能与圆弧相连，圆弧既可以与直线相连，也可以与圆弧相连，但都应保证连接点处两元素的所在直线或切线在同一直线上。系统在这方面未加限制，需要用户自行控制。

对于一个新建的赛道，在未进行任何操作之前，赛道起点，即坐标点 $(0, 0)$ ，位于当前主视窗的中心处。可以将鼠标移动到主视窗的中心处，查看左下角的状态栏显示的坐标是否是 $(0, 0)$ 。

了解系统关于赛道设计的规定之后，就可以进行赛道的设计了。

【添加直线元素】

只需在直线元素设计栏中输入终点坐标，点击“直线”按钮，即完成了直线元素的添加与绘制。

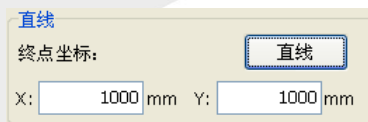


图 3.2.5 直线元素设计栏

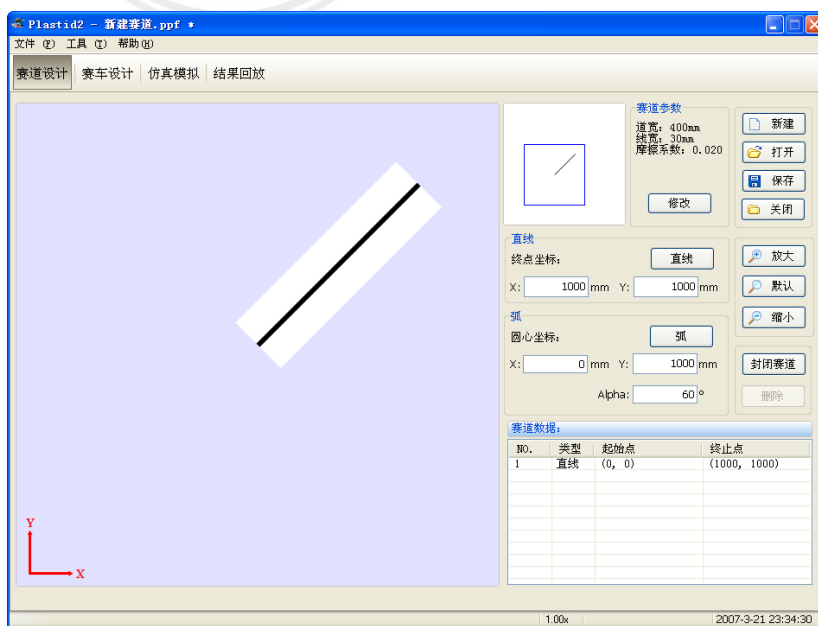


图 3.2.6 添加直线元素

同时，赛道数据栏中会按绘制顺序列出当前赛道所有赛道元素，包括元素的顺序编号，起始点、终止点坐标。

赛道数据:			
NO.	类型	起始点	终止点
1	直线	(0, 0)	(1000, 1000)

图 3.2.7 赛道数据栏

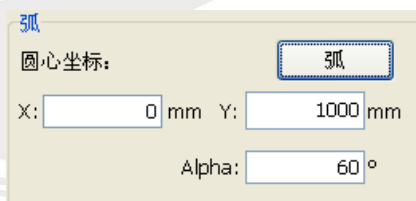
当输入的直线元素终点坐标与起始点坐标相同时，该输入为无效值，系统将给出错误警告。



图 3.2.8 直线元素错误警告

【添加弧元素】

在起始点确定的情况下，弧元素只需要圆心坐标和起始点到终止点扫过的角度即可确定。在弧元素设计栏中输入圆心坐标和扫过角度 Alpha 之后，点击“弧”按钮，即完成弧元素的添加与绘制。



弧

圆心坐标:

X: mm Y: mm

Alpha: °

图 3.2.9 弧元素设计栏

同样的，新添加的弧元素也会添加到赛道数据列表中。

此时，新添加的弧元素已经有部分超出了主视窗外，为了能够观察弧元素的剩下部分，将鼠标移动到主视窗，按下鼠标左键，拖动鼠标，即可移动主视窗，等到需要观察的弧元素完全出现在主视窗内之后，松开鼠标左键，即完成了主视窗的移动。

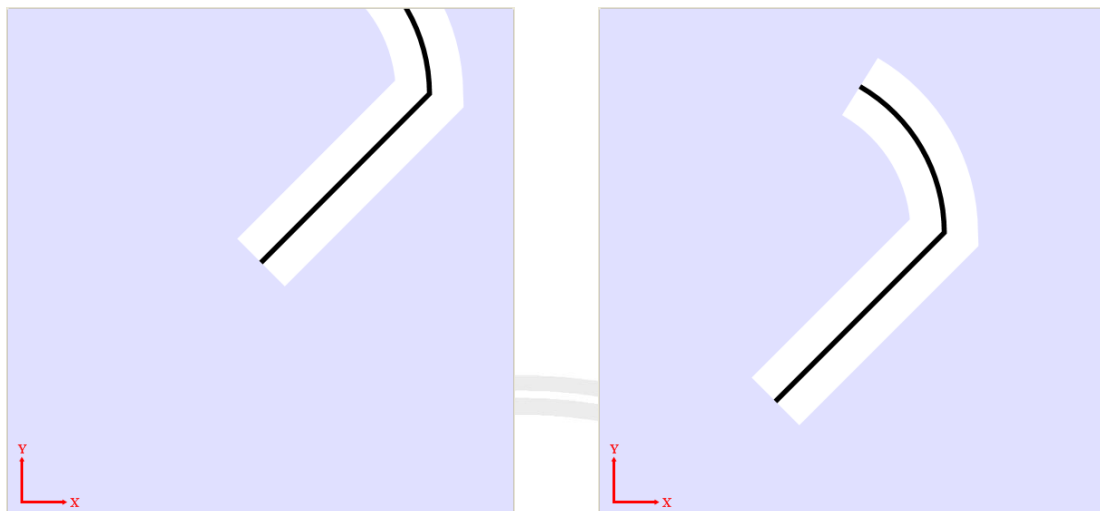


图 3.2.10 移动主视窗：移动前（左），移动后（右）

当输入的弧元素圆心坐标与起始点坐标相同时，该输入为无效值，系统将给出错误警告。



图 3.2.11 弧元素错误警告

比赛规则中，有关于赛道最小半径的相关规定，系统会根据设置的规则设定^①检查弧元素的半径，如果半径小于规则设定，则给出错误提示。

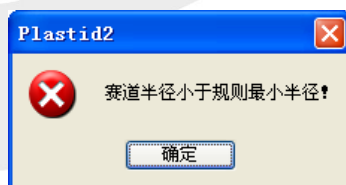


图 3.2.12 规则检查错误警告

【封闭赛道】

添加完一系列赛道元素之后，如果要设计封闭赛道的话，就需要回到赛道的起始点。点击“封闭赛道”按钮，系统自动添加一条最后添加元素的终止点到赛道起始点的直线元素，将赛道封闭起来。

^① 赛道规则的设定在“默认参数和规则设置”章节详细介绍。

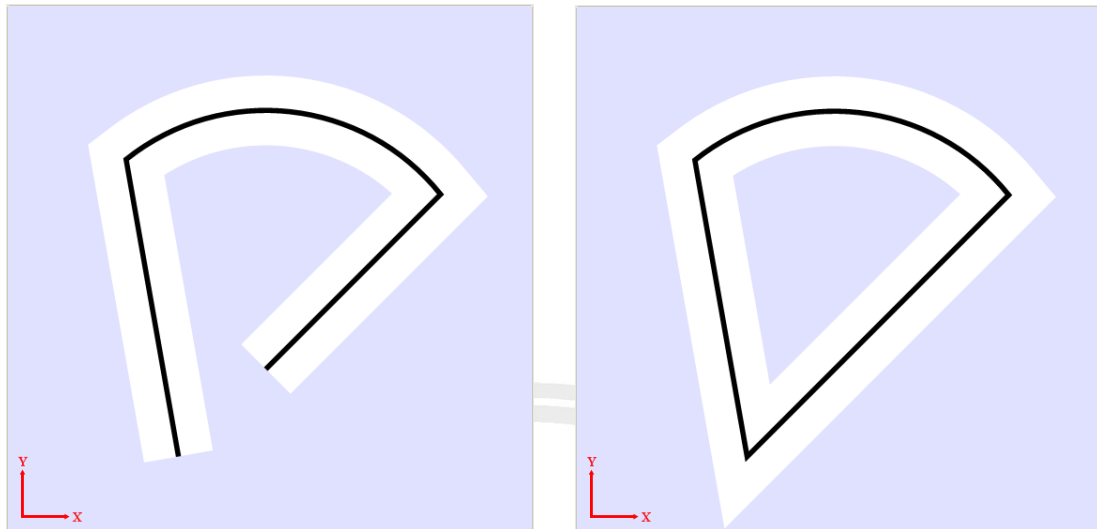


图 3.2.13 封闭赛道：封闭前（左），封闭后（右）

【删除赛道元素】

如果对设计的赛道不满意，可以删除其中不满意的元素。删除赛道元素时，是将当前选择的赛道元素及之后的所有元素一并删除。选择了赛道元素之后，点击“删除”按钮，在弹出的确认对话框中选择“是”，将完成删除过程。

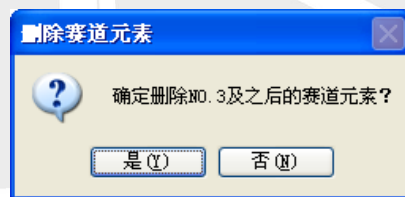


图 3.2.14 删除赛道元素确认对话框

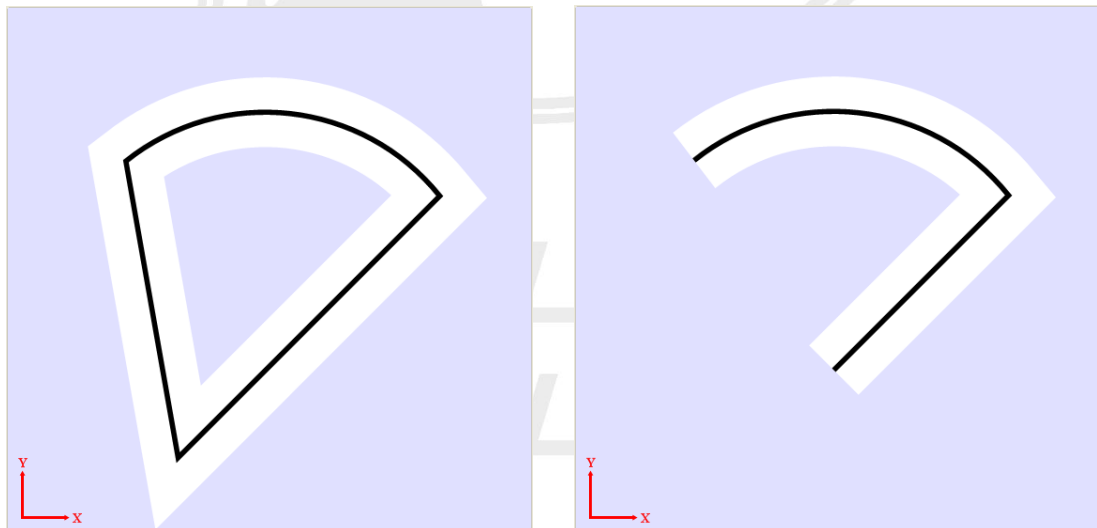


图 3.2.15 删除赛道元素：删除前（左），删除后（右）

【缩放赛道】

有时候为了更仔细的观察局部或者查看更大范围的局部，就需要用到缩放功能。



图 3.2.16 赛道缩放栏

点击“放大”按钮，将会以当前主视窗中心点所对应的坐标点为基准点对赛道局部进行放大显示，相应的全局视窗中表示观察范围的蓝色矩形框将变小，表示观察范围的缩小；点击“缩小”按钮，将会对赛道局部进行缩小显示，蓝色矩形框将变大，表示观察范围的扩大；点击“默认”按钮，将恢复到系统默认的缩放比例进行显示。放大、缩小操作均可连续进行，直到达到用户理想的效果。

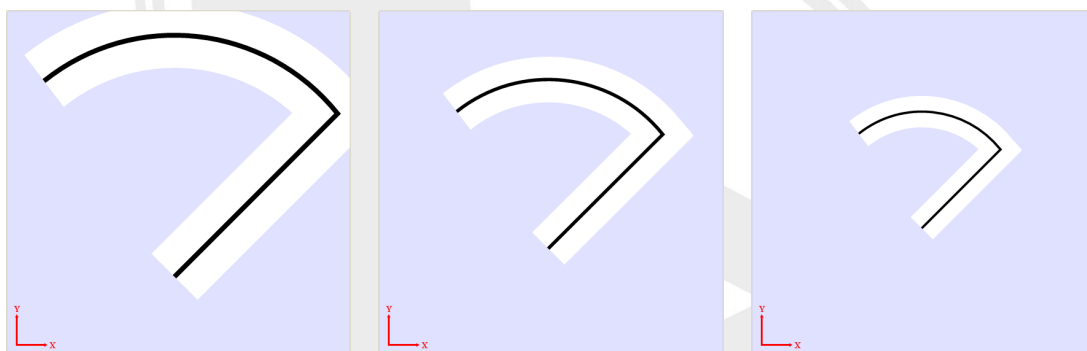


图 3.2.17 赛道缩放效果：放大（左），默认（中），缩小（右）

【修改赛道参数】

赛道参数包括赛道宽度、中心线宽度和摩擦系数三项，显示在赛道参数栏中。



图 3.2.18 赛道参数栏

各赛道参数代表的意义如下：

摩擦系数：小车在赛道上行驶时的摩擦系数；

赛道宽度：直线赛道（圆弧段）中非连接处的赛道的宽度；

中心线宽度：赛道中心的循迹线的宽度，即为小车提供循迹信号的中心线的宽度。

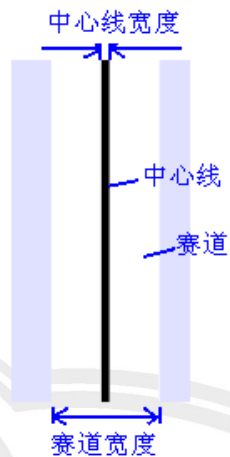


图 3.2.19 赛道结构

新建赛道时，赛道参数为默认值^①，如果用户需要修改赛道参数，点击赛道参数栏的“修改”按钮，在弹出的赛道参数修改对话框中将赛道参数修改为希望值，点击“确定”，即可完成修改。如果修改了赛道宽度和中心线宽度，则主视窗中会反映出修改所发生的变化。

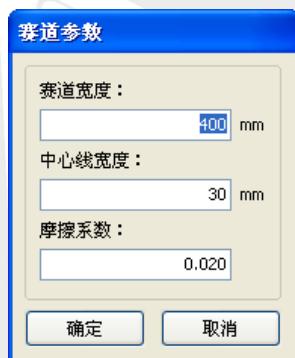


图 3.2.19 赛道参数修改对话框

^① 赛道参数默认值的设定在“默认参数和规则设置”章节详细介绍。

第四章 赛车设计

4.1 功能介绍

赛车模型的设计，可以说是整个仿真系统最重要的一环。赛车模型设计的好坏，将直接影响到仿真过程中各种状态量的计算，最终影响到仿真结果对控制算法验证的可信度。因此，设计一个好的赛车模型，可以使仿真结果更好的逼近实际情况，为控制算法提供更好可信度的验证，从而提高开发效率。

在开始赛车模型设计的介绍之前，先明确一下系统关于赛车模型设计的一些设定：

赛车模型参数的值必须在具有物理意义的范围内，如车重必须大于 0g，车长必须大于 0mm，电机的额定电压必须大于 0V 等。系统会对设定的值进行检查，如果不在物理意义范围内，系统将认为该值无效，并将该值设置为最近接设定值的物理意义范围内的值。

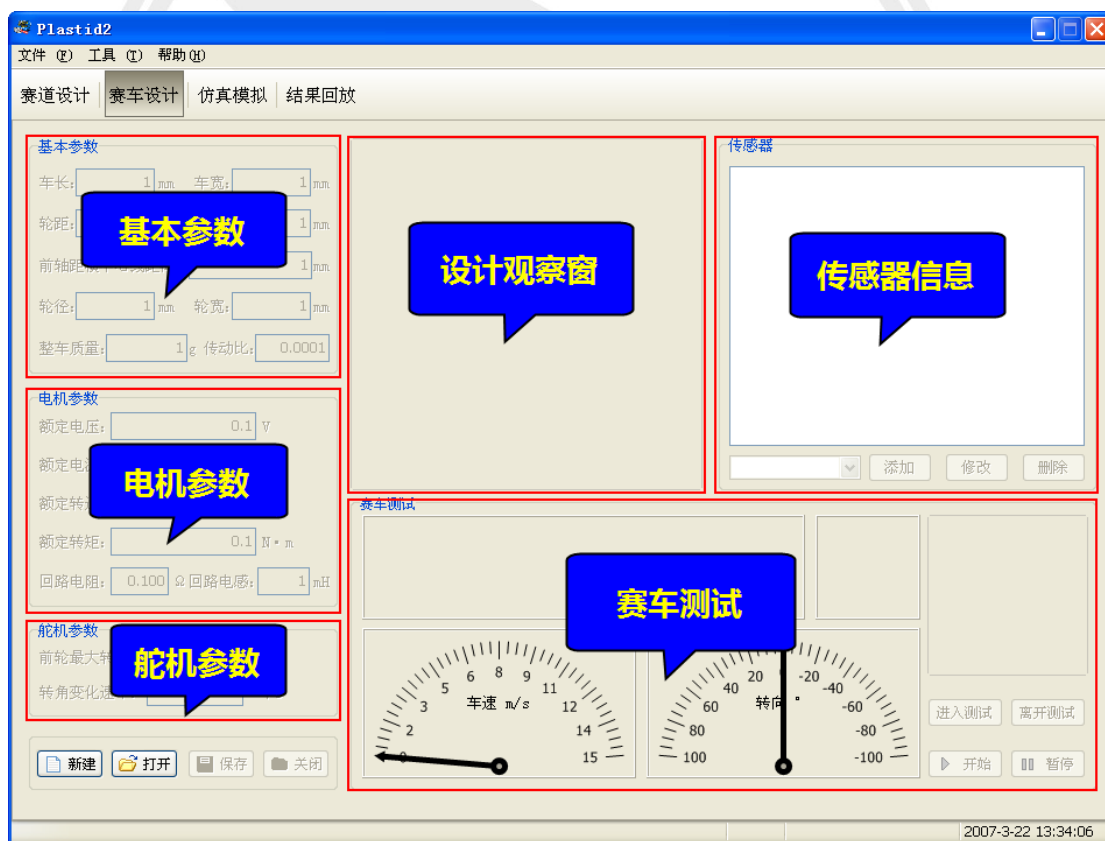


图 4.1 赛车设计界面

4.2 设计赛车

4.2.1 赛车文件操作

【新建赛车】

与新建赛道类似，点击“新建”，程序将以默认参数^①新建一个初始赛车，初始赛车仅包括赛车的基本参数、电机参数和舵机参数，不包括传感器数据。

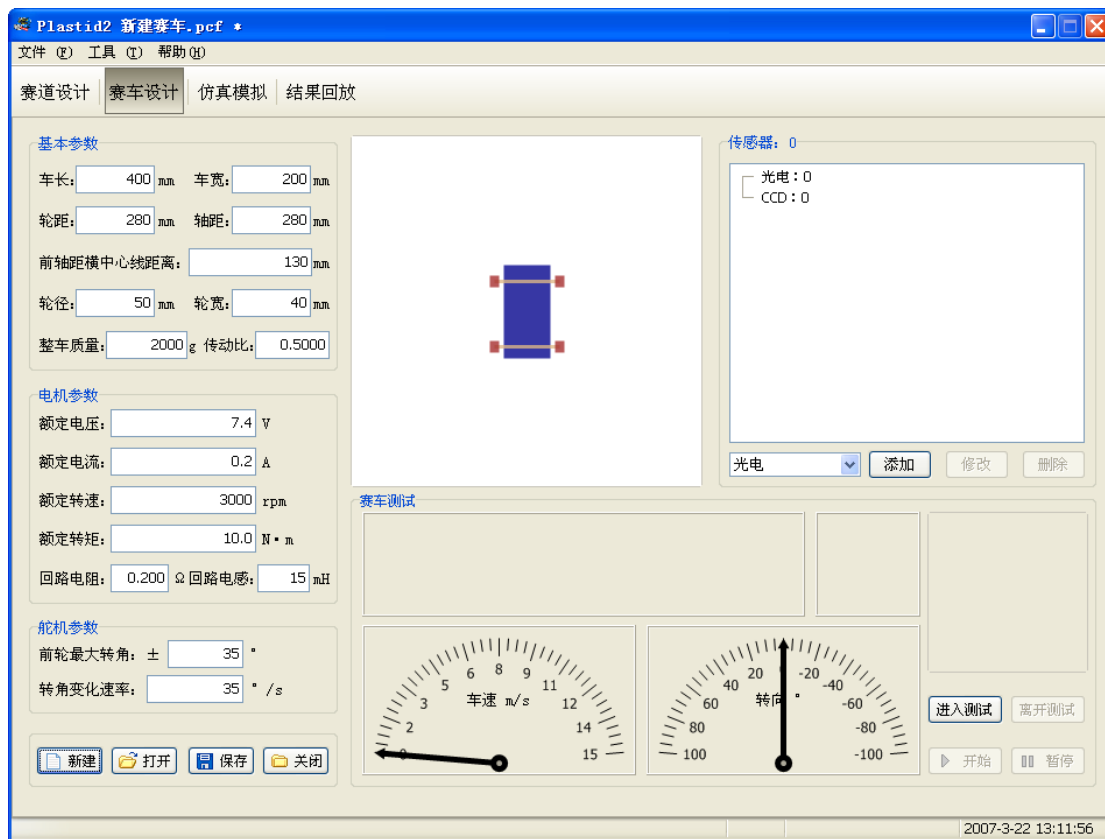


图 4.2.1 新建赛车

【打开已有赛车】

与打开已有赛道类似，Plastid2 中提供有示例赛车，用户刚开始使用时可以打开提供的示例文件。

点击“打开”，在弹出的赛车文件选择对话框中选择需要打开的赛车文件，程序将加载该赛车文件。



提示：程序 Car 文件夹下的 CCD.pcf 和 Ligth.pcf 为程序提供的示例赛车文件，其中 CCD.pcf 为带有 CCD 传感器的赛车模型，Ligth.pcf 为带有光电传感器的赛车模型。Plastid2 的赛车道文件后缀名为 pcf。

^① 赛车参数默认值的设定在“默认参数和规则设置”章节详细介绍。

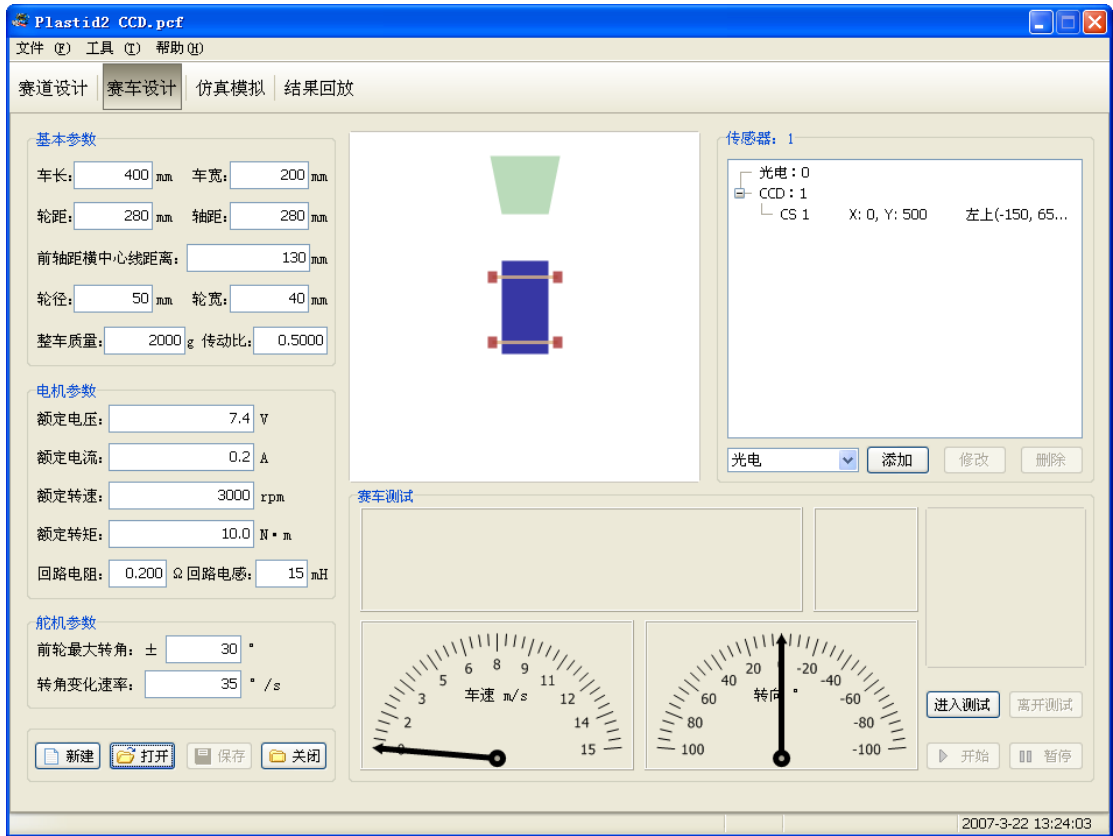


图 4.2.2 打开已有赛车

【保存、关闭赛车】

具体过程与赛道设计类似，在此不再赘述。

4.2.2 赛车设计

赛车设计包括：赛车基本参数设置、电机参数设置、舵机参数设置、添加并设置传感器参数四个方面。获得并设置准确的各种参数，是一个好的赛车模型的基础。

【赛车参数】

首先是赛车基本参数，也即赛车的机械参数。

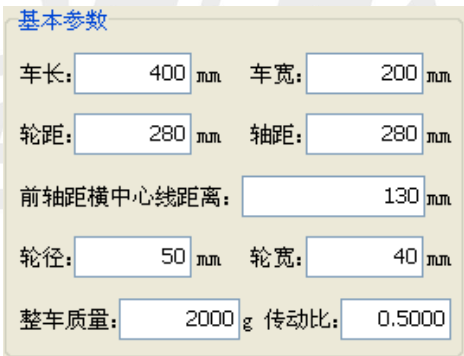


图 4.2.3 赛车基本参数栏

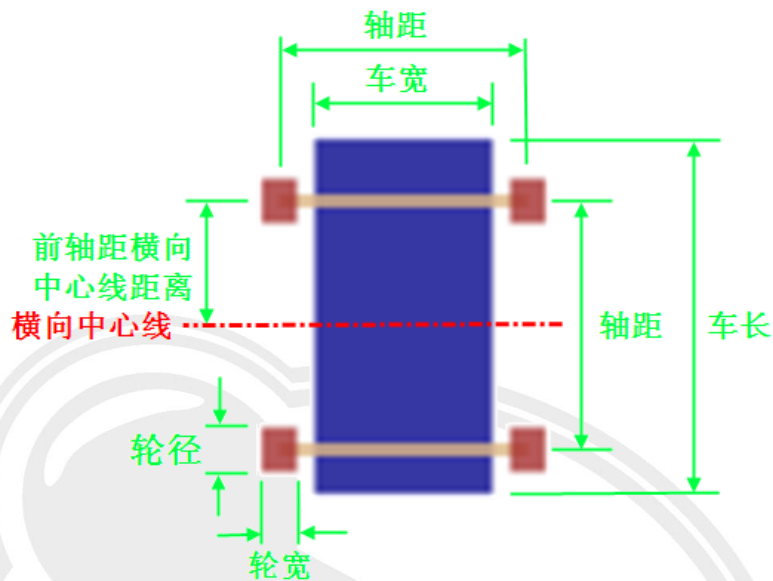


图 4.2.4 赛车外形参数

车长、车宽、轮距、轴距、前轴距横向中心线距离、轮径、轮宽：赛车的外形参数，单位均为 mm，具体物理意义见上图。对赛车外形参数的修改，将即时的在设计观察窗中显示出来。

整车质量：小车装上电池、单片机开发板、传感器以及相应的驱动电路之后的总质量，单位为 g。

传动比：电机输出端到车轮处的传动比，其值为车轮转速 / 电机转速。

其次，是电机参数，即赛车的加速参数。

额定电压、电流、转速、转矩等参数都是电机已知的参数，这需要用户根据自己电机的参数来设定，Plastid2 的电机模型采用的是闭环直流调速电机，关于电机模型的具体细节，可以参见《电机与运动控制系统》（杨耕 罗应立等编著 清华大学出版社），这里就对其原理不作描述。

这里需要提及的是关于回路电阻和电感（电枢回路）参数的设置，关于这两个参数的设置，我们推荐用户采用压降法来进行测量：在电枢回路中通入接近额定值的电流，用低量程电压表测量正负电刷间的压降，再除以电枢电流即为电枢回路电阻，同样的方法可以测量得到电感，这需要用户根据自己设计的电机驱动电路来进行测量。

最后，是舵机参数，即赛车的转向参数。

前轮最大转角：赛车前轮能够转过的最大角度。

转角变化速率：车轮转动过程中转角变化的速率。

在实际测试中发现，赛车前轮转角与舵机控制的 PWM 成线性关系，并且其变化速率近似为固定值，因此舵机模型只设置这两个参数。同时，这两个参数不但与舵机有关，也与赛车舵机到前轴的机械传动结构有关，需要用户实际测试获得。

【添加传感器】

传感器选择和布置是整个比赛中对控制算法影响最大的一个因素。良好的传感器选择和布置，可以提供更详细准确的赛道信息，同时也可以使算法更加合理和优化。所以只有在良好的传感器选择和布置的前提下，决策程序才有可能发挥出它应有的作用。

1) 添加光电传感器

从第一届的比赛来看，采用光电传感器的队伍占了参赛队伍的大部分，因为光电传感器的使用相对来说比较简单。当然，这只是对于简单的使用光电传感器的信号进行非连续识别的应用来说，如果使用光电传感器的信号进行非连续识别，那就是比较高阶的应用，应用起来也不是那么简单的。



图 4.2.5 传感器信息栏

在传感器栏信息中，在“添加”按钮前的下拉列表中选择“光电”，然后点击“添加”。在弹出的光电传感器对话框中，选择光电传感器的类型并根据需要输入传感器工作有效半径，输入传感器安装参数之后，点击“确定”，即可完成光电传感器的添加。

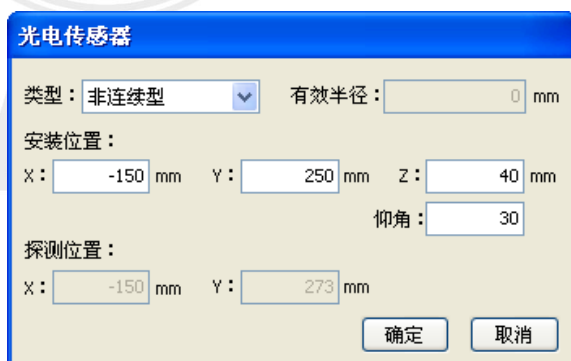


图 4.2.6 光电传感器对话框

系统关于传感器安装位置的一些约定：

以用户为参考，用户看向车的方向为俯视。传感器安装位置的参考坐标系，X 轴与赛车的横向中心线重合，向右为正；Y 轴与赛车的纵向中心线重合，向上为正；原点为 X 轴与 Y 轴交点；Z 轴垂直于 XY 平面，指向用户为正。仰角为为了探测更远的距离，光电传感器的

安装基板的上仰角度（与水平线的夹角）。同样的，添加 CCD 传感器时的俯角为安装 CCD 时的下俯角度。

光电传感器的工作类型，分为连续型与非连续型两种。光电传感器在工作过程中，其输出值为一个模拟量。对于比赛中的黑色轨迹线来说，光电传感器的输出值与传感器探测点（光电发射管的照射中心点）到黑色轨迹线的中心线的距离有关：当该距离大于某一有效值 L 时，输出值为最大值；距离为 0 时，输出值为 0；当该距离在 $0 \sim L$ 之间时，输出值随 L 线性变化。

因此，光电传感器模型分别提供了连续型和非连续型两种类型供用户选择。如果选择非连续型，则传感器探测点在黑色轨迹线上时，输出为 0，不在时为 255^①；如果选择连续型，则有效半径 R 即为前述的 L ，当传感器探测点到赛道黑色轨迹线的中心线的距离大于 R 时，输出值为 255；在 $0 \sim R$ 之间时，输出值在 $0 \sim 255$ 之间线性变化。

光电传感器的添加结果也即时的在设计观察窗中显示出来，相应的其信息也被添加到传感器数据树状表中，如图 4.2.7 所示，设计观察窗中的绿色圆点表示添加的光电传感器。如果传感器树状列表中显示的传感器参数信息不完全，请将鼠标移动到想要查看的传感器上，在鼠标下方弹出的提示窗口中将完全显示树状图中未能完全显示的参数信息。传感器数据树状表中显示的是传感器的实际探测位置等参数信息，如果需要查看详细的参数，请选择想要查看的传感器，然后点击“修改”，在弹出的对话框中查看。

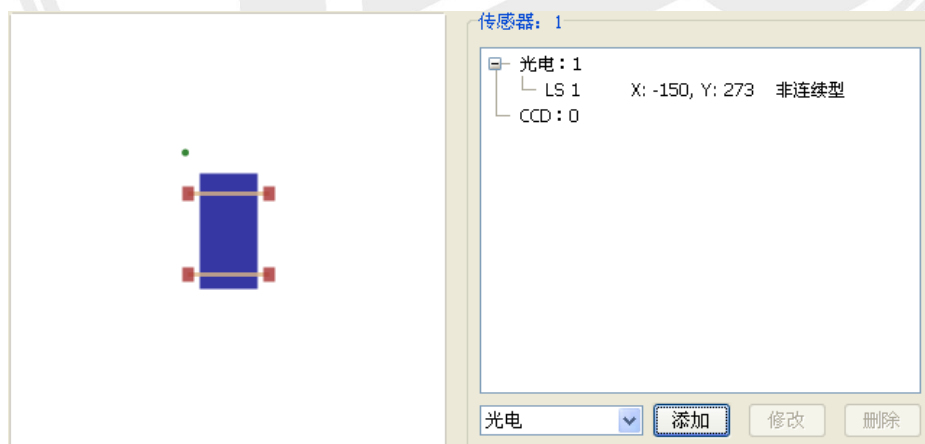


图 4.2.7 光电传感器添加结果

2) 添加 CCD 传感器

这里的 CCD 传感器泛指摄像头，包括了 CMOS、CCD 等不同工艺。从使用上来说，CCD 传感器比起光电传感器要复杂的多，但相应的能够提供的信息也更多，可以应用许多更高级的算法。

在传感器栏信息中，在“添加”按钮前的下拉列表中选择“CCD”，然后点击“添加”。在弹出的 CCD 传感器对话框中，输入传感器安装参数、像素值、探测位置等参数之后，点击“确定”，即可完成 CCD 传感器的添加。

^① 对应于 S12 上的 8bit AD 转换。



CCD传感器

CCD像素：

横向：100 Pix 纵向：100 Pix

安装位置：

X：0 mm Y：0 mm Z：350 mm

俯角：35°

探测位置：

X：0 mm Y：500 mm

感应范围：

左上角：X：-150 mm Y：650 mm

左下角：X：-100 mm Y：400 mm

右上角：X：150 mm Y：650 mm

右下角：X：100 mm Y：400 mm

确定 取消

图 4.2.8 CCD 传感器对话框

CCD 像素为实际控制算法中使用的 CCD 传感器像素值。

安装位置参数同光电传感器。

探测位置为通过安装位置 4 个参数确定的 CCD 传感器中心点所探测的位置，下面的感应范围参数的设置将受到探测位置的限制，如左上角的 X 不能大于探测位置的 X，Y 不能小于探测位置的 Y。

感应范围为 CCD 传感器感应的赛道范围，其参考坐标系同安装位置。感应范围需要用户自行对所使用的 CCD 传感器进行标定获得。



提示：CCD 像素值并不是越多越好。因为 S12 的运行频率仅为 24MHz，如果使用图 4.2.8 中的 100*100 的像素值，那么每帧数据就是 10000 个点，CCD 发送数据的速率是 24 帧，光对这些点的处理就将占用 S12 绝大部分的处理能力。同时，使用太大的 CCD 像素值，也将严重拖慢仿真时的速度。

同样的，CCD 传感器的添加结果也将即时的在设计观察窗中显示出来，相应的参数信息也被添加到传感器数据树状表中，如图 4.2.9 所示。图中的绿色梯形框为 CCD 传感器的探测范围。

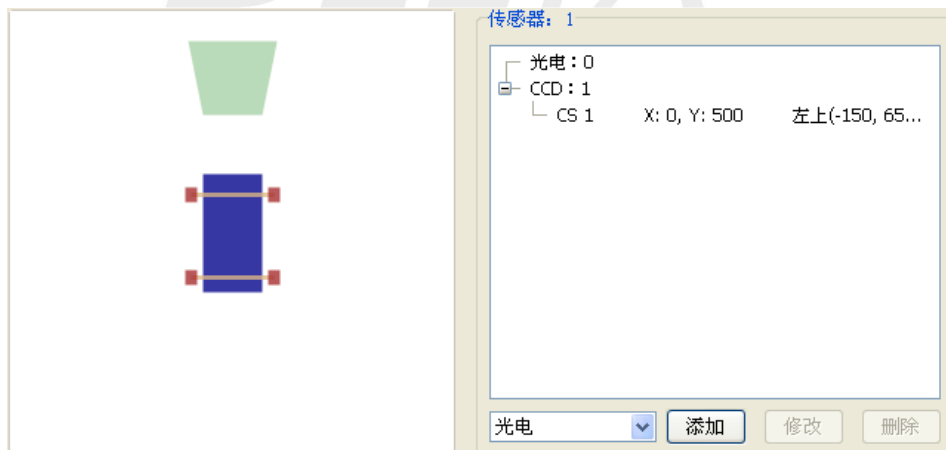


图 4.2.9 光电传感器添加结果

【修改、删除传感器】

如果想修改传感器的参数，先在传感器树状列表中选择需要修改的传感器，然后点击“修改”，在弹出的光电传感器对话框中进行修改，点击“确定”，即完成修改。

如果想删除传感器，先在传感器树状列表中选择想要的传感器，然后点击“删除”，在弹出的删除传感器确认框中点击“是”，即完成删除。



DELTA
STUDIO

4.3 测试赛车

4.3.1 功能介绍

当设置完赛车的设计之后，用户可以通过 Plastid2 提供的测试功能对设计的小车进行简单的测试，主要是通过对小车的手动操作来观察小车的性能和传感器的布置是否合适。

点击界面右下方的“进入测试”，系统将会进入测试模块。此时，赛车设计界面中的所有参数将会被锁定，用户只能对小车进行运动操作，直到用户退出测试，赛车设计界面中的参数才会解除锁定，用户方可对其进行修改。

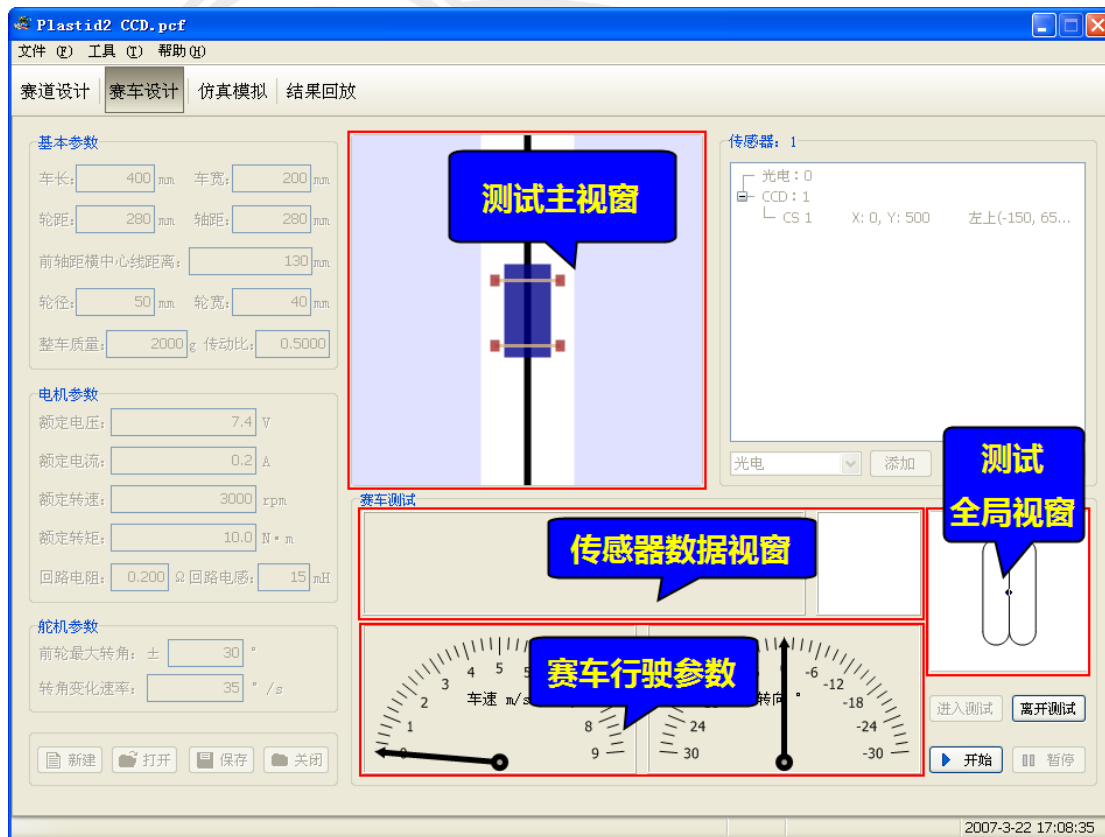


图 4.3.1 赛车测试界面

测试主视窗：即为设计赛车的设计观察窗，显示当前赛车的行驶姿态。


传感器视窗：用于显示赛车行驶过程中的传感器数据。左侧为光电传感器视窗，一个绿色的小圆圈代表一个光电传感器，其相对位置与光电传感器在赛车上的布置相同。圆圈内的填充颜色反映对应传感器的当前输出值，对于非连续型光电传感器，只有白色和墨绿色两种颜色；对于连续型传感器，则在白色和墨绿色之间变化。

测试全局视窗：显示测试赛道全局视图，并显示赛车当前在赛道上的位置。

赛车行驶参数视窗：显示赛车当前的行驶车速和前轮转角。

系统关于赛车前轮转角的约定如下：

前轮未发生偏转，与车身平行时，转角为 0° ；前轮向左偏转时，转角为正；前轮向右偏转时，转角为负。

 提示：测试时自动载入的赛道文件为程序目录下的 Default.ppf，如果对测试赛道不满意，可以自行设计新赛道，覆盖该文件即可。

4.3.2 测试操作

进入测试之后，点击“开始”为开始进行测试；点击“离开测试”为退出测试，解除赛车设计界面中对赛车参数的锁定。

开始测试后，键盘的“上”、“下”键分别控制赛车的加减速；“左”、“右”键控制赛车前轮的转向。

测试过程中，点击“暂停”，暂停当前测试，再次点击“开始”，则继续当前测试。



DELTA
STUDIO

第五章 仿真模拟

5.1 功能介绍

Plastid2 的核心功能就是仿真模拟，仿真环境通过调用用户设计好的赛道、赛车、传感器等模型，对用户编写的控制算法进行仿真模拟。由于系统已经对整个实际的运行环境已经有了一个尽可能近似的描述，仿真出来的结果已经可以对用户的控制算法进行一个合理的评价。

点击上方的“仿真模拟”功能标签，即进入仿真模拟子界面。

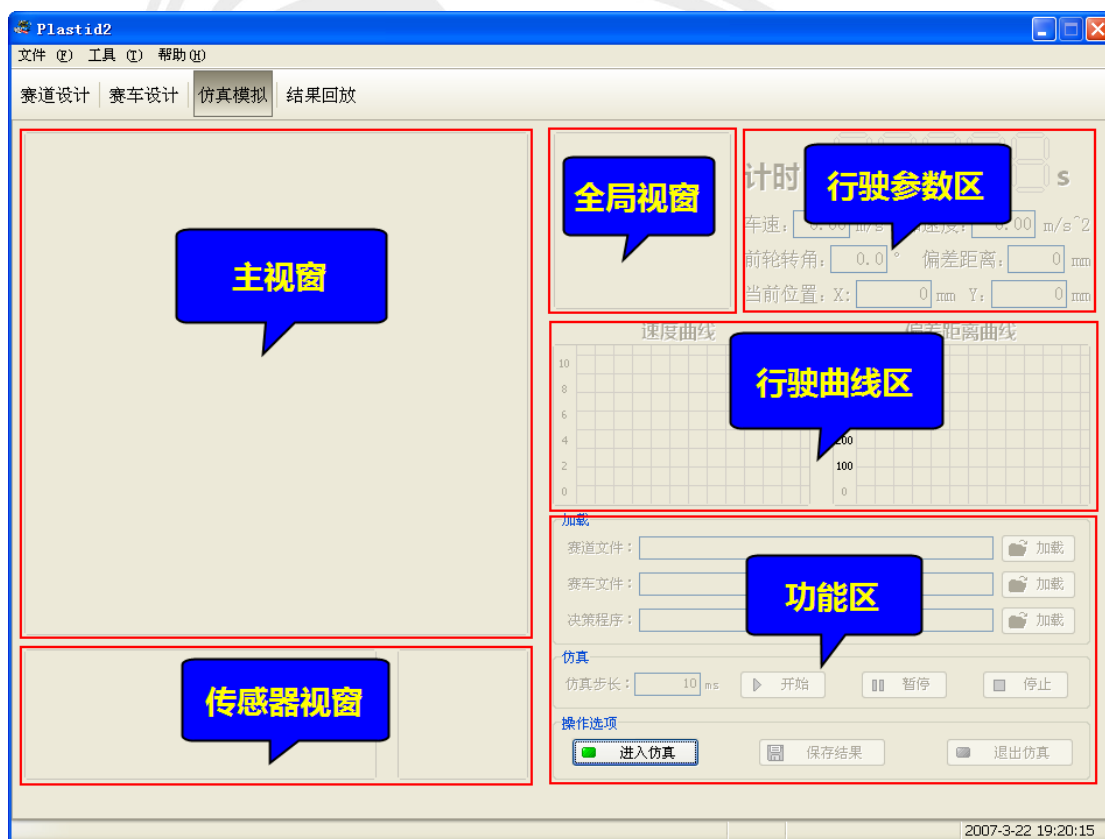


图 5.1.1 仿真模拟子界面

5.2 仿真模拟

点击“进入仿真”，将进入仿真模块，右侧的行驶参数区、行驶曲线区、功能区相应功能按钮解除锁定；点击“退出仿真”，将退出仿真模块，相应模块重新恢复锁定状态。

5.2.1 加载

仿真环境需要同时具备赛道模型、赛车模型和控制算法才能进行仿真，因此，在开始仿真前，需要先分别加载赛道文件、赛车文件和控制算法文件。

在功能区的加载栏，点击右侧的三个“加载”按钮，可以分别浏览选定赛道文件、赛车文件和控制算法文件进行加载。如果加载正确，将在相应的文本栏内显示加载的文件名，如下图所示。

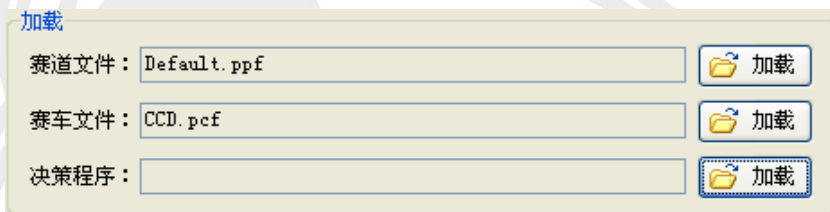


图 5.2.1 加载栏

加载的赛道文件和赛车文件就是前面的“赛道设计”和“赛车设计”两个模块所生成的文件。控制算法文件是用户编写的 dll 文件。关于控制算法文件的编写，请参见附录 A。

所有文件加载完毕之后，仿真的准备工作就已经完成了，效果如下图。

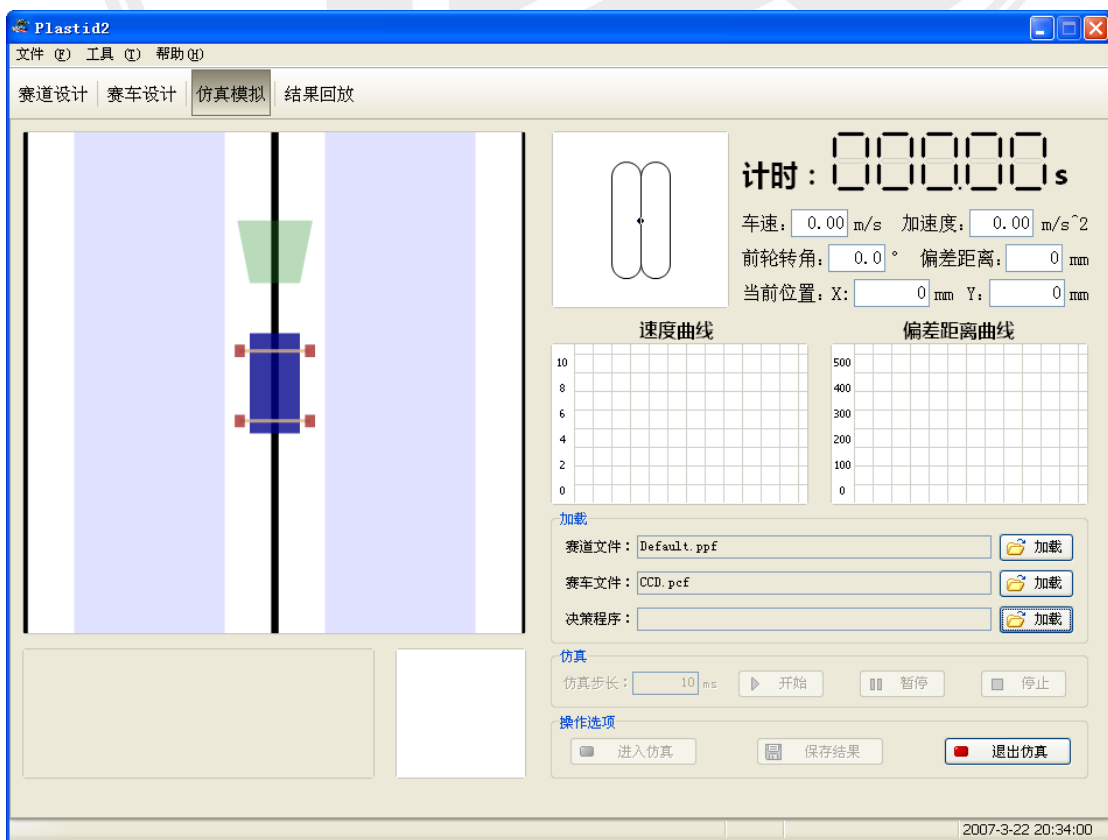


图 5.2.2 加载完毕

5.2.2 仿真模拟

赛道、赛车、控制算法文件加载完毕之后，设定好仿真步长，即可开始仿真。

仿真步长：每次仿真计算的时间间隔。因为计算机是离散系统，要对赛车行驶过程进行仿真模拟，就需要对该过程进行离散处理，仿真步长即相当于采样时间间隔。实际上仿真步长也是控制算法的决策周期。系统允许的最小仿真步长为 10ms。

点击“开始”，即开始仿真。系统提供了“暂停”和“停止”两个功能，其功能都是停止当前仿真，但是“暂停”只是暂时将仿真停止在当前状态，再次点击“开始”之后，仿真又可以继续进行，而“停止”则是停止仿真，并将仿真开始时的初始状态，再次点击“开始”，将从初始状态重新开始进行仿真。

当“暂停”或者“停止”仿真之后，可以点击“保存结果”对仿真数据进行保存，以供回放分析时调用。点击“退出仿真”将退出仿真模块。



提示：Plastid2 的赛道文件后缀名为 prf。

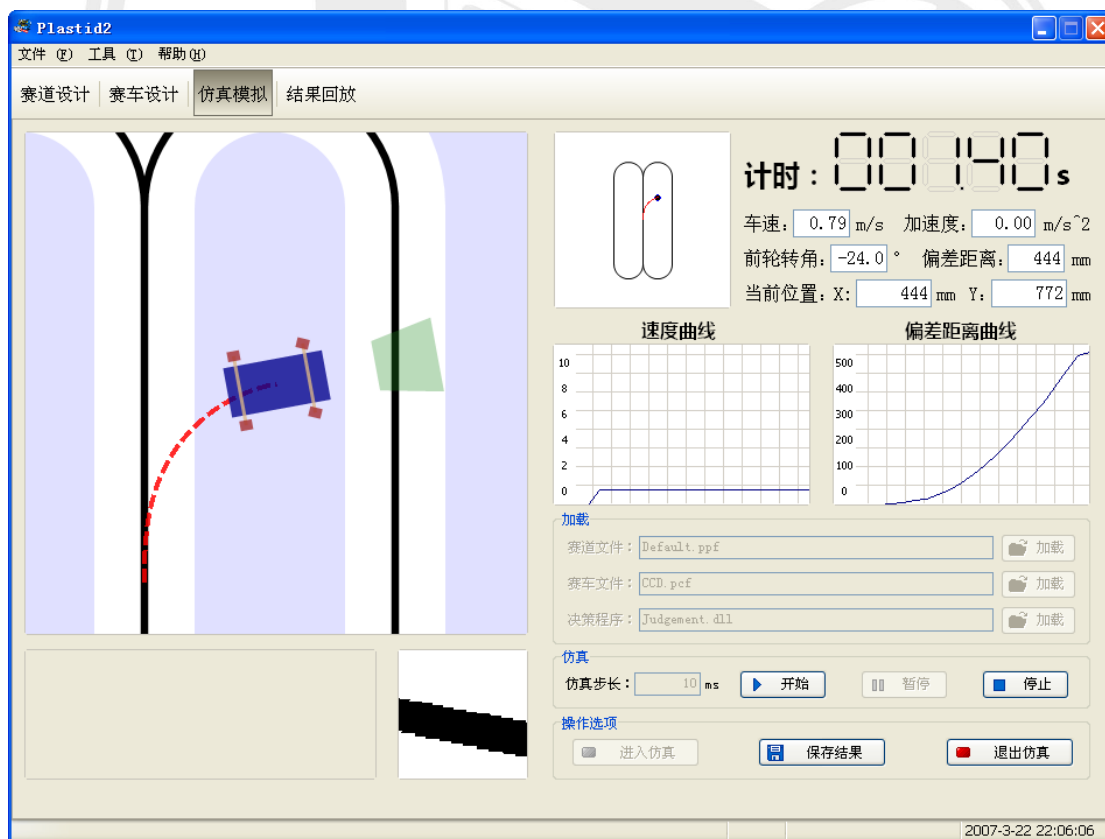


图 5.2.3 仿真模拟过程

仿真开始之后，系统提供了 5 块区域显示赛车行驶过程中的各种状态参数：

1) 主视窗

显示局部区域的赛车运行姿态，用于细节观察。

2) 全局视窗

显示全局信息，用于观察赛车运行全局状态。

3) 传感器视窗

显示传感器信号，同“赛车测试”。


4) 行驶参数区

显示当前时间，赛车车速、加速度、前轮转角、偏差距离、位置等状态参数。

偏差距离：赛车中心点到赛道中心线的最短距离。

5) 行驶曲线区

显示速度和偏差距离的历史曲线，用于观察两者变化的趋势。是评估控制算法效果的重要依据。

 提示：如果觉得前面赛车设计子界面对赛车或赛道的测试太过简单，没能仔细观察整个过程，可以在加载控制算法文件的时候，选择程序 **Judgement** 文件夹下的 **KeyBoard.dll** 进行加载，开始仿真模拟后，与赛车测试过程类似，通过方向键上下左右控制赛车的加减速和转向。

DELTA
STUDIO

第六章 结果回放

6.1 功能介绍

一个好的控制算法总是需要经过长时间的调试才能总结出来,所以对每次仿真保存的结果进行分析是算法调试的一个重要环节。“结果回放”模块可以对仿真保存的结果进行回放,并提供多种控制功能,使用户能够按照自己希望的方式控制回放过程,方便清晰地观察、分析、比较仿真的结果。

点击上方的“结果回放”功能标签,即进入结果回放子界面。

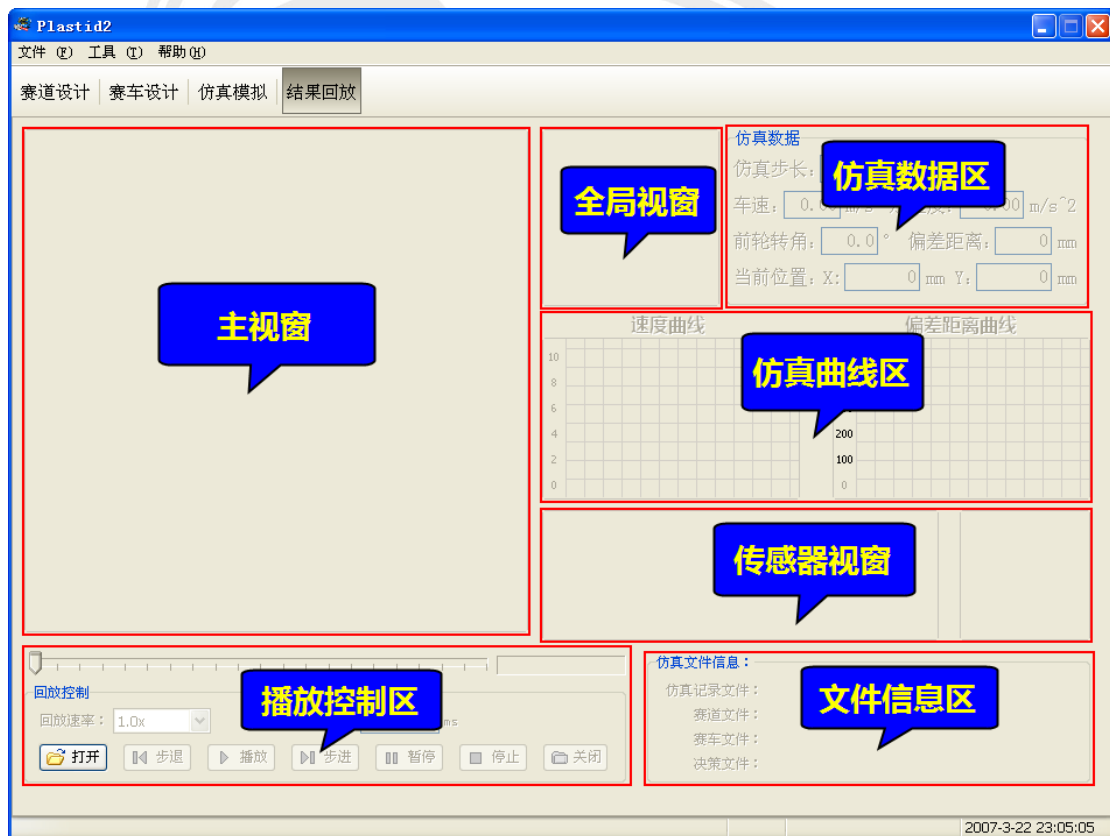


图 6.1.1 结果回放子界面

6.2 结果回放

6.2.1 加载仿真结果文件

进入结果回放子界面后，界面中除了“打开”按钮之外，所有元素都处于锁定状态。点击“打开”，在弹出的仿真记录文件选择对话框中选择需要打开的仿真记录文件，程序将加载该仿真记录文件。文件加载成功后，结果回放子界面的将解锁并显示相应信息。点击“关闭”将关闭当前仿真记录文件，重新锁定界面元素。

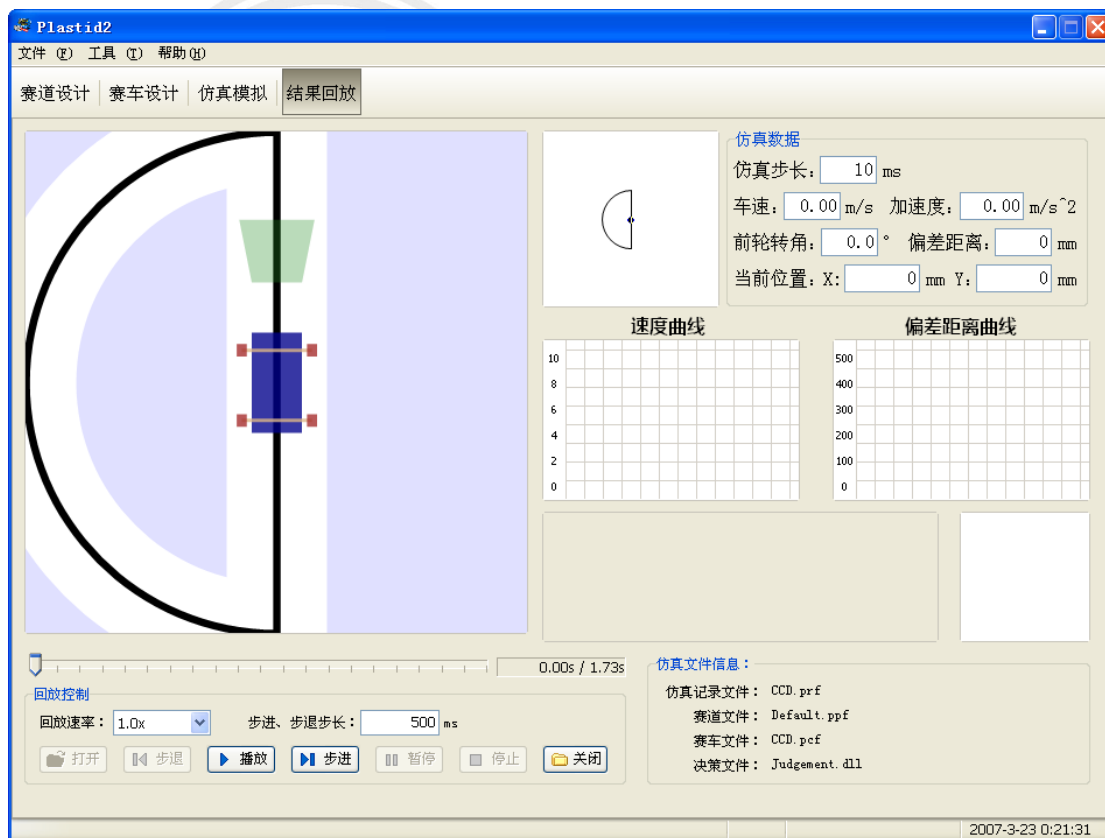


图 6.2.1 仿真记录加载完毕

1) 主视窗

同仿真模拟子界面的主视窗，显示局部区域的赛车运行姿态，用于细节观察。

2) 全局视窗

同仿真模拟子界面的全局视窗，显示全局信息，用于观察赛车运行全局状态。

3) 传感器视窗

同仿真模拟子界面的传感器视窗，显示传感器信号。

4) 仿真数据区

同仿真模拟子界面的行驶参数区，显示仿真步长，赛车车速、加速度、前轮转角、偏差距离、位置等状态参数。

5) 仿真曲线区

同仿真模拟子界面的行驶曲线区，显示速度和偏差距离的历史曲线，用于观察两者变化

的趋势。

6) 文件信息区

显示当前仿真记录文件名，仿真时加载的赛道、赛车和控制算法文件名。

6.2.2 回放控制

回放控制功能实现集中在回放控制区。仿真文件加载后，点击“播放”即可开始回放仿真结果。

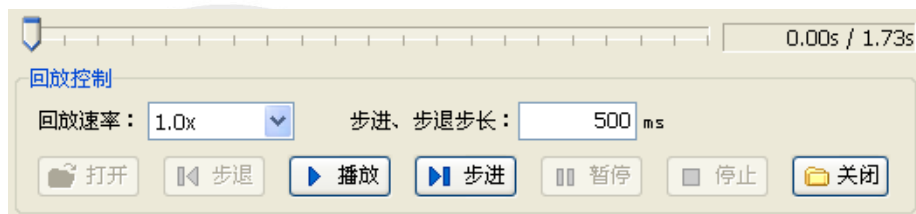


图 6.2.2 回放控制区

回放控制区上方的进度条显示当前播放进度，用鼠标拖动进度条可以实时调整播放进度。进度条右侧为时间栏，显示仿真记录文件总时间和当前回访时间。

在回放过程中，可以随时“暂停”或“停止”回放。

回放速率可调，在回放速率下拉列表中，提供了 0.1x、0.2x、0.4x、0.6x、0.8x、1.0x6 档播放速率以供选择。

步进、步退功能，即向后或向前跳过指定时间，该时间由“步进、步退步长”栏中设定的数值决定。系统约定：步进、步退步长不得小于仿真步长，否则，将以仿真步长为步进、步退步长。步进、步退功能，不论是在播放还是暂停、停止状态，都可以进行步进、步退操作。

以上回放控制功能的组合，使得用户能够自由的控制回放过程，更好的对仿真结果进行分析。

DELTA
STUDIO

第七章 默认参数与规则设置

7.1 默认参数设置

赛道与赛车模型，都有大量的必需参数，如果每次新建赛道或赛车时都需要对这些必需参数进行逐一设置，将是一件低效繁琐的事情。对于每次比赛来说，赛道和赛车的必需参数基本都是固定的，那么，将这些值设为默认参数，每次新建赛道或赛车时都用这些默认值来初始化必需参数，用户不需要或者只需少量改动，将有效避免这种低效繁琐的劳动，从而提高设计时的效率。

在主菜单的“工具”栏中选择“选项”，弹出选项对话框。

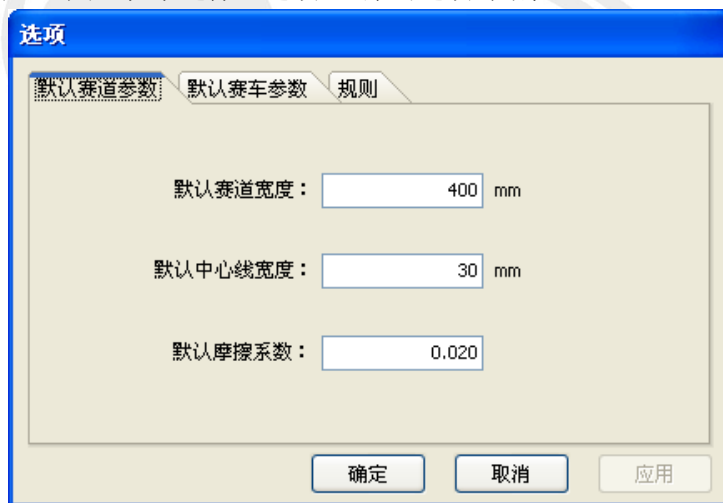


图 7.1.1 默认赛道参数标签

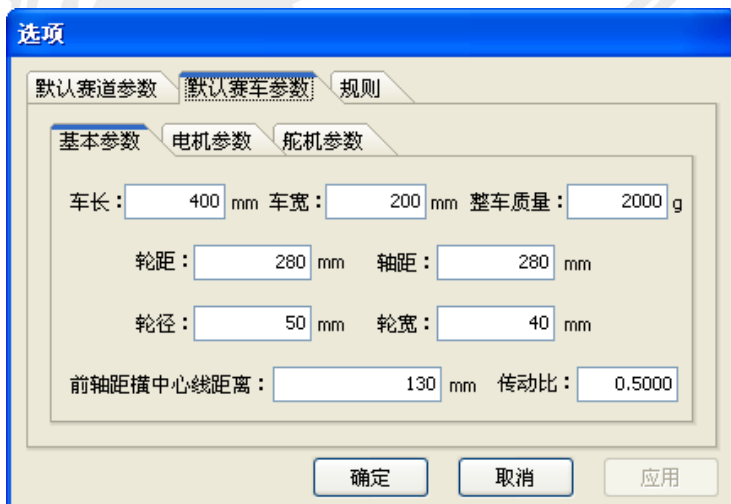


图 7.1.2 默认赛车参数标签

在选项对话框的“默认赛道参数”和“默认赛车参数”标签中，可以分别设置赛道和赛车默认参数。

7.2 规则设置

在比赛规则中,有关于最大传感器数量和赛道最小半径的相关规定。在选项对话框的“规则”标签栏中可以进行相应的设置。设置完成后,系统将在赛道设计和赛车设计过程中进行相应的检查,如果超出规则允许,将给出错误提示。

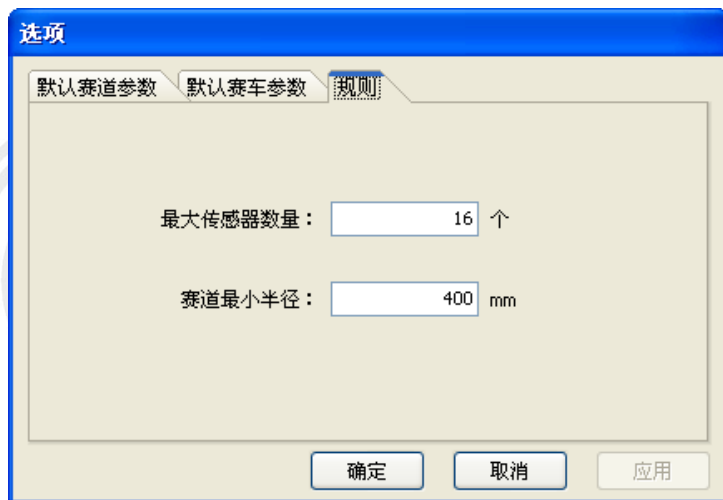


图 7.2.1 规则标签

DELTA
STUDIO

附录 A 如何编写控制算法文件

Plastid2 通过调用控制算法文件中的控制算法函数，完成仿真模拟过程。控制算法文件是 Windows 下通用的 dll 文件，其中包含 Judge 和 Reset 两个必需函数，可以使用任何支持 windows 的编程环境编写。

Plastid2 在程序的 Project 文件加下提供了 VC（VC6、VC2005）和 Delphi（Delphi7、Delphi2006）控制算法文件模板，使用 VC 或 Delphi 编写控制算法文件时可以直接使用，如果使用 VB 等其它编程环境请自行参考编写。

VC 下的函数声明：

```
extern "C" __declspec(dllexport)
void Judge( unsigned short Speed,
            unsigned char* SensorData, unsigned short SensorCount,
            unsigned char* CCDDData, int CCDWidth, int CCDHeight,
            unsigned short* MotorPWM, unsigned short* SteerPWM){}
```

```
extern "C" __declspec(dllexport)
void Reset(){}
```

Delphi 下的函数声明：

```
procedure Judge(Speed: Word;
                SensorData: PChar; SensorCount: Word;
                CCDDData: PChar; CCDWidth: Integer; CCDHeight: Integer;
                MotorPWM: PWord; SteerPWM: PWord); stdcall;

begin
end;

procedure Reset();
begin
end;
```

各函数及其参数意义如下：

1) Judge 函数

控制算法决策函数，Plastid2 通过该函数输入赛车、传感器状态参数，获得电机、舵机控制量。

Speed: 电机当前转速。

SensorData、SensorCount: SensorData 为光电传感器输出值数组。当光电传感器为非

连续型时，0 表示探测点在赛道黑色轨迹线上，255 表示探测点不在赛道黑色轨迹线上；为连续型时，255 表示传感器到赛道黑色轨迹线的中心线的距离 L 不小于有效半径 R ，0~255 之间时， $L = \text{传感器输出值} * R / 255$ 。SesorCount 为数组长度，用于防止数组的越界访问。

CCDData、CCDWidth、CCDHeight: SensorData 为 CCD 传感器输出值数组，255 表示该像素点为白色，0 表示该像素点为黑色。CCDWidth 和 CCDHeight 分别为 CCD 的横向像素值和纵向像素值，同样是用于防止数组的越界访问。


MotorPWM、SteerPWM: 用于输出决策程序的控制量，MotorPWM 用于输出电机控制 PWM，SteerPWM 用于输出舵机控制 PWM。

输出的控制 PWM 对应到实际 PWM 的换算公式为 $PWM = \text{MotorPWM} (\text{SteerPWM}) / 65535$ 。

2) Reset 函数

重置函数，用于进行控制算法的重置，如仿真停止后重新开始，PID 调节参数中累计误差的清零等。

在提供的控制算法模板中，已经有简单的控制代码，演示了 Reset 函数的使用以及电机、舵机 PWM 输出。编译成功后即可在仿真模拟中调用。

 提示：如果在仿真模拟中遇到加载控制算法文件错误的提示，请检查控制算法文件的编写，是否出现了越界访问等错误。

DELTA
STUDIO

附录 B 如何导出仿真数据

Plastid2 已经提供了强大的数据回放功能，可以灵活控制回放的整个过程，并有详细的数据显示。但是，在编写一些更高级的算法时，有可能对仿真数据的处理有更高的要求。对此，Plastid2 提供了专门的仿真数据导出工具，可将数据导出以供更高级的分析处理使用。

双击运行程序目录下的 **Export.exe** 文件。



图 B.1 仿真数据导出工具

点击“浏览”，在弹出的文件选择对话框中，选择需要导出的仿真数据记录文件，导出工具会显示出选择的数据记录文件的路径信息以及文件中数据的一些简要信息。



图 B.2 选定仿真数据记录文件

点击“导出”，在弹出的文件保存对话框中输入想要保存的文件名，点击“保存”，即完成数据导出过程。导出的数据文件为 **tab** 分隔的文本文件，可以用 **Excel** 打开，其格式如下：

时间 (ms)	坐标 X	坐标 Y	速度 (m/s)	加速度 (m/s^2)	偏差距离 (mm)	前轮转角 (°)	光电传感器 输出值
0	0.0	0.0	0.00	0.00	0.0	0.0	FF,0,FF

可以导出的传感器数据仅限于光电传感器，其值为十六进制表示，各传感器值之间用，分隔。