Chapter 1

Introduction

Welcome to the world of C-STORM (Computer Simulation Tool for the Optimization of Robot Motions). C-STORM is a software package designed to make it easy to model, simulate and generate optimal motions for a wide range of mechanisms. This manual is an attempt to provide new users with enough information to begin using the software. Since C-STORM has been integrated into Matlab, this manual assumes the reader has some knowledge of Matlab (at least enough to generate simple .m files). For more information about Matlab, see the Matlab User Manual.

C-STORM has evolved from several students work on robot path optimization. These students often had widely different backgrounds, goals and abilities. As such, the software can often seem perplexing. For example, several methods may exist to accomplish the same goal – each reflecting a different student’s efforts. In addition, there are often “left over” or stray bits of code from past students work which no longer work as originally designed. While this is not a problem for someone familiar with these eccentricities, it can be very frustrating for the new user. This manual attempts to describe the current functionality of the software. It is important to note that C-STORM is in a constant state of flux as students add new abilities. As I am not always aware of this work, it may be left out of this manual. My hope is that any significant new work done on the software will be accompanied by addendums to this manual.
1.1 General software description

As mentioned earlier, C-STORM evolved from a C++ software program. C++ provided fast execution speeds and a powerful programming language. However, it required the user to have a working knowledge of C++. This requirement was a major stumbling block for many new users. This greatly curtailed the use of the software. To eliminate this problem and provide a wider user base, we decided to integrate the software package into Matlab. Matlab provided a more user friendly scripted program language as well as several of the optimization tools needed by the software. Matlab’s scripted language did not provide the execution speed needed, but by using *mex* files, we were able to continue using a large portion of the existing C++ code. This allowed us the best of both worlds: the user friendly and familiar interface of Matlab and the execution speed of compiled C++ code.

While integrating the software package into Matlab, we made extensive use of the new class structures of Matlab 5.x. This allowed us to duplicate the existing C++ classes inside Matlab. In Chapter 8, you will find a complete description of the Matlab classes, *m* files, *mex* files and C++ code.

Instead of presenting all of these descriptions now, I will instead concentrate on formulating and solving a typical problem. This will allow us to introduce the necessary tools at a more leisurely pace. It will also allow us to separate those functions which are necessary from those which are merely internal book-keeping and can be ignored by the typical user.
Chapter 2

Installing the C-STORM package

C-STORM is fairly easy to install. It is normally distributed in a file called cstorm.zip for the PC and it should be accompanied by a readme.txt file which outlines the installation procedure.

1. Extract cstorm.zip into a directory of your choosing. Make sure to retain the directory structure embedded in the .zip file.

2. Add the directory to the Matlab path

3. Add the subdirectory called disp.gui to the Matlab path

Once this has been done you are ready to begin using the C-STORM package. You may need to restart Matlab to make sure it accepted the new path directories.