SoftWalls in 2D

J. Adam Cataldo, Edward A. Lee, Xiaojun Liu, Ashwin Ganesan,
and Stephen Neuendorffer
{acataldo, eal, liuxj, ganesan, neuendor}@eecs.berkeley.edu
Electrical Engineering & Computer Science
University of California, Berkeley

25 June 2002

1 Introduction

This document gives an outline of a general 2D softwalls approach. We assume the initial speed is known.

2 Step Bias

Suppose the aircraft is travelling at a constant speed s. Let θ denote the aircraft's heading angle, and let $\dot{\theta_p}$ be the pilot's control input, i.e. the desired rate of change in heading angle. At that speed, the aircraft has a minimum-safe turning radius r_{min}. This contrains the paths at which the aircraft can bank right or left as in figure 1.

Now suppose we wish to prevent the aircraft from crossing a boundary, or entering a no-fly zone. As the aircraft approaches the boundary, one of minimum-turning-radius paths from figure 1 will interesect with the boundary. As long as the other path has not yet interestected the boundary, the aircraft can still avoid crossing the boundary by moving along this path. At the instant the second path intersects the boundary, forcing the aircraft along the second path will prevent the aircraft from crossing the boundary. This situation is depicted in figure2.

We can now pose this control algorithm in a more general control framework. Let the actual rate of change in aircraft heading angle be$\dot{\theta}$, and let θ_s, be the control signal generated by the SoftWalls algorithm. Given the pilot's control signal,$\dot{\theta_p}$, we calculate $\dot{\theta}$ from

$$\dot{\theta} = limit_{[-s/r_{min}, s/r_{min}]}(\dot{\theta_p} - \dot{\theta}_s),$$

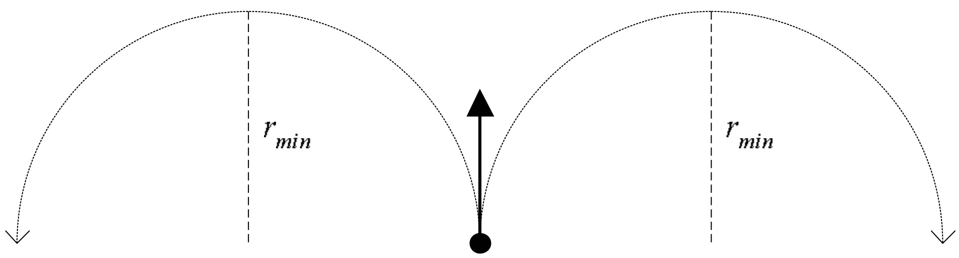

Figure 1: The dot represents the aircraft, which is moving along the center path. The arc paths representent turns at the minimum turning radius.

where

$$limit_{[a,b]}(u) = \begin{cases} b & \text{if } u > b, \\ a & \text{if } u < a, \\ u & \text{otherwise.} \end{cases}$$

Assuming $\dot{\theta}_p$ is limited to $[-\dot{\theta_M}, \dot{\theta_M}]$, which corresponds to the limits of the flight yoke, we can choose $\dot{\theta}_s = \dot{\theta_M} + s/r_{min}$ to force the aircraft to turn right at the maximum turning radius, irrespective of pilot input. Similarly$\dot{\theta}_s = -\dot{\theta_M} - s/r_{min}$ will force the aircraft to turn left. If the left path is the second path to interesect the boundary, the control signal $\dot{\theta}_s = \dot{\theta_M} + s/r_{min}$ will steer the aircraft right, and if the right path is the second to interestect the boundrary, the control signal $\dot{\theta}_s = -\dot{\theta_M} - s/r_{min}$ will steer the aircraft left. By applying the appropriate θ_s to the input, the softwalls algorithm (the path/boundary-interestection algorithm) can garauntee the plane to never crosses the boundary.

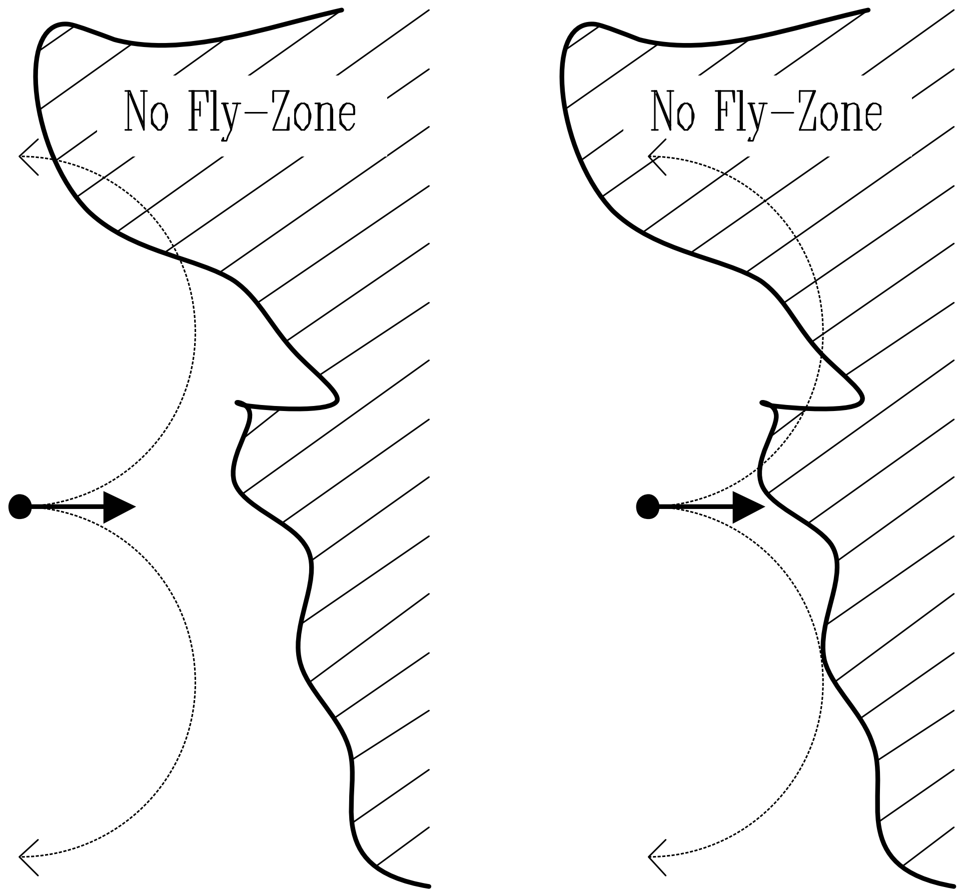

Figure 2: On the left picture, the aircraft still has one free path. On the right picture, the aircraft must move right at the minimum turning radius to avoid crossing the boundary.