2nd International Conference on Autonomous Robots and Agents
December 13-15, 2004 Palmerston North, New Zealand

# Obstacle Avoidance and Finite-Time Tracking of Mobile Targets

Daniel J. Stonier
Faculty of Informatics and Communication,
Central Queensland University,
Rockhampton, Queensland, Australia.

Russel J. Stonier
Faculty of Informatics and Communication,
Central Queensland University,
Rockhampton, Queensland, Australia.

## Abstract

In this paper we consider the finite time control of a simulated point mass robot with acceleration control to a target in an environment possessing obstacles that the robot must avoid. Both target and obstacles may be non-stationary. The algorithm presented combines classical Liapunov techniques and Terminal Sliding Modes to simultaneously achieve obstacle avoidance and finite time convergence - properties previously distinct to both methods.

**Keywords**: Finite time control, mobile robot, obstacle avoidance, terminal sliding mode.

## 1 Introduction

Obstacle avoidance techniques have existed for classical mathematical methods using Liapunov functions for some time [1]. The various methods have different merits depending on the circumstances involved, however they all have one thing in common - convergence to their desired target point is only asymptotic.

In contrast, a technique known as *Terminal Sliding Control* [2, 3] enables convergence to the desired target in finite time, but did not previously allow for any form of directional control - thus obstacle avoidance was not possible.

This paper continues the initial progress made in [4], [5] in coupling both techniques to provide a control which achieves both finite convergence and obstacle avoidance. In [4], a technique using a slightly modified terminal sliding control was used to achieve both a high degree of boundary avoidance and a continued assurance of finite time convergence to the target. The problems involving obstacle avoidance on a more general scale were discussed.

In [5], a classical gradient method was coupled with the terminal sliding mode to provide directional control of the object to its target. This allowed for full obstacle avoidance as well as maintaining finite time convergence to the target. The control to the target was in general much quicker as it provided a more intelligent path to the desired target. Discontinuities in the first derivative of the sliding mode however, prevented an analysis involving mobile targets and obstacles.

Here we remove the problem discontinuities and simplify the sliding mode task to focus only on the radial convergence of the object to its target (thus sliding mode is only 1-dimensional). Again, we couple the method with a classical gradient technique to choose the direction of travel required (other methods may be used - we use a steepest descent method here for its simplicity of illustration). Once the direction is ascertained, the sliding mode parameters are adjusted to ensure the path is tracked. The analysis is also developed to incorporate mobile targets and obstacles. Several examples are included at the end for illustrative purposes.

## 2 Model Formulation

We consider a single point mass object (robot) at position $(x,y)$ and its target at $(t_x,t_y)$ (as measured on a fixed cartesian frame). The target may be non-stationary and is free to move within the boundaries of the system.

We will also refer to the robot's position relative to its target using polar notation (refer to Figure 1).

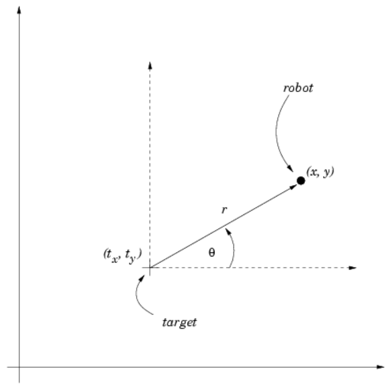


Figure 1: Co-ordinate Definitions

The dynamics of the point mass object is then given by the equations

$$\ddot{r} = u_r \qquad\qquad \ddot{\theta} = u_\theta. \tag{1}$$

where $(u_r, u_\theta)$ represent radial and angular *relative* accelerations. For practical purposes these may be easily transformed back to the original fixed Cartesian frame.

No restrictions are imposed on the controls $(u_r, u_\theta)$ - the purpose here is to illustrate the theory behind the algorithm only. It can be observed however that the controls calculated in the examples to follow are relatively minimal in magnitude.

# 3 The Control Problem

## 3.1 The Liapunov Function

We propose the Liapunov function, S (Sliding Liapunov Function),

$$\begin{aligned} S &= \frac{1}{2}\left[(\dot{r} + \beta_r r^{m/n})^2 + (\dot{\theta} - \beta_\theta)^2\right], \\ &= \frac{1}{2}\left[S_1^2 + S_2^2\right]. \end{aligned} \tag{2}$$

where

- $S_1$ – Terminal Sliding Component
- $S_2$ – Angular Tracking Component
- $\beta_r$ – Terminal Sliding Parameter
- $\beta_\theta$ – Angular Tracking Parameter

$S_1$ is the usual form of a 1-dimensional terminal sliding function. $S_2$ is not required for convergence to the target here, but nonetheless needs to be controlled for directional tracking and obstacle avoidance (discussed in 5). Parameters $\beta_r$ and $\beta_\theta$ are time varying parameters. The constants $m, n$ are chosen such that $n > m$ and $m, n$ are positive integers.

The control problem is to find controls that reduce $S$ to zero in finite time.

At this point the radial component will be said in **sliding mode** (obeying dynamics specified by $\dot{r} + \beta_r r^{m/n} = 0$) and the angular component will be in **angular tracking mode** ($\dot{\theta} = \beta_\theta$).

To determine these controls, first note that $S$ is a well-defined and positive definite function on $\mathbb{R} \times [0, 2\pi]$. The derivative exists everywhere except at $r = 0$ and is of the form

$$\begin{aligned} \dot{S} = S_1 &\left(u_r + \dot{\beta}_r r^{m/n} + \frac{m}{n}\beta_r r^{(m-n)/n}\dot{r}\right) \\ &+ S_2\left(u_\theta + \dot{\beta}_\theta\right). \end{aligned} \tag{3}$$

## 3.2 The Control

If we choose controls

$$\begin{aligned} u_r &= -K\mathrm{sgn}(S_1) - \dot{\beta}_r r^{m/n} - \frac{m}{n}\beta_r r^{(m-n)/n}\dot{r}, \\ u_\theta &= -K\mathrm{sgn}(S_2) - \dot{\beta}_\theta, \end{aligned} \tag{4}$$

for some $K > 0$. Then

$$\dot{S} = -K(|S_1| + |S_2|) \leq -K|S|^{1/2}, \tag{5}$$

and convergence to $S = 0$ will occur in finite time.

Once convergence occurs ($S = S_1 = S_2 = 0$), we will say the system is in **tracking mode**.

## 3.3 Resolving the Discontinuity

There exists a possible discontinuity in the control as $r \rightarrow 0$. Since $r = 0$ is only approached along a sliding mode, we may observe that for a general choice of $m, n$

$$\dot{r} = -\beta_r r^{m/n}, \tag{6}$$

while sliding and that the component in $u_r$

$$r^{(m-n)/n}\dot{r} \rightarrow -\beta_r r^{(m-n)/n} r^{m/n}, \tag{7}$$

which tends to the value $-\beta_r r^{(2m-n)/n}$. Consequently there will be a singularity in $u_r$ unless $m, n$ is chosen so that $2m > n$. To satisfy this requirement we set $m = 3$ and $n = 5$ for the examples to follow.

By only requiring a single sliding mode as compared to the Cartesian-based modes in [5] we have removed the discontinuities that occurred if initial placement of the robots coincided with the target axes.

# 4 Directionless Tracking

For both this and the following section we will work under the assumption that the controls have already forced the system into *tracking mode* (that is, $S_1 = S_2 = 0$). As was shown in the previous section, the controls will ensure the system remains in *tracking mode*.

For an *obstacle free environment* we set $\beta_\theta = 0$ (no directional tracking necessary, simply direct convergence to the target) and adopt the usual approach for terminal sliding problems by defining $\beta_r$ as a positive constant.

Motion resulting as the robot slides to the target may be found by solving the separable differential equation

$$S_1 = \dot{r} + \beta_r r^{m/n} = 0. \tag{8}$$

This results in the integration,

$$\int_{r_0}^{r(t)} \frac{dr}{r^{m/n}} = -\int_0^t \frac{dt}{\beta_r}, \tag{9}$$

where $r_0$ is the radial displacement from the target upon commencement of sliding. Solving the above integral for convergence to the target, that is $r(t_f) = 0$ where $t_f$ is the convergence time, it can be seen the robot converges to the target in finite time where

$$t_f = \left(\frac{n-m}{n}\right) \beta_r r_0^{(n-m)/n}.$$

## 5 Directional Tracking

Consider now an environment configured with boundaries and obstacles of varying sizes. The target is located at $(t_x, t_y)$ (possibly mobile) and once again we assume that the controls have already ensured the robot in tracking mode. To choose a direction suitable for the robot to follow, we construct a Liapunov function of the form [5]

$$L(x,y) = \frac{1}{2}\left[(x-t_x)^2 + (y-t_y)^2\right]\left[\frac{\lambda_1}{V_1} + \frac{\lambda_2}{V_2} + \ldots\right]$$

where $V_i$ represents the magnitude of the distance between the robot and the i-th obstacle and the $\lambda_i$ represent positive penalty multiplier constants that prioritise the importance of each obstacle.

Given the position of the robot at some point $(x,y)$, some examples of the distance functions $V_i$ used in the simulations that follow are of the form

**Boundaries** (Straight Wall) - A wall at $x = 3$ may be represented by the distance function

$$V_1 = |x-3|. \tag{10}$$

Equivalently, we may also use the function

$$V_2 = (x-3)^2. \tag{11}$$

**Obstacles** (Circular Disc) - A suitable distance function for a circular disc loacted at $(2,2)$ with radius 0.5 would take the form

$$V_3 = \sqrt{(x-2)^2 + (y-2)^2} - 0.5. \tag{12}$$

Essentially, the Liapunov function $L$ simply maps the local terrain in terms of mountains and valleys, with a global minimum (valley) located at the target. From this the direction of steepest descent is calculated and it is the path traced out by the direction of steepest descent that provides the guide for sliding and tracking modes.

Navigating along a path of steepest descent has well-known flaws. Such an environment may find the path trapped in local valleys or saddle points for which there may be no unique direction of steepest descent. However, other more robust and complicated methods may be used in its place to find a suitable path, and since the purpose of the paper here is to show that directional control (with sliding modes) with finite time convergence is achievable, we will use this method (steepest descent) for its simplicity in illustration.

If we assume an environment free of the problems outlined above, the direction of steepest descent is given by the unit vector

$$\hat{\underset{\sim}{l}} = -\underset{\sim}{\nabla} L / |\underset{\sim}{\nabla} L|. \tag{13}$$

Converting to a vector representation defined in the relative polar co-ordinate frame, we use the notation

$$\hat{\underset{\sim}{l}} = l_r \underset{\sim}{e}_r + l_\theta \underset{\sim}{e}_\theta. \tag{14}$$

This subsequently represents the desired direction of travel required. We then define our $\beta_r$, $\beta_\theta$ as the parameters given by

$$\beta_r = -l_r \qquad\qquad \beta_\theta = l_\theta. \tag{15}$$

Note that these parameters will now be time varying (changing as the robot and its target move). By choosing a time varying sliding parameter we diverge from the usual approach used for terminal sliding modes.

Typically the path taken by an object converging to its target in a sliding mode problem is fixed once the object starts sliding. However by varying the $\beta$ we can manipulate the sliding mode in such a fashion so that it traces a path of our choosing. This is what provides the directional control.

Mathematically, the most important difference lies in the fact that the *sliding parameter* $\beta_r$ *may no longer always be positive* and consequently finite time convergence, or terminal sliding, is not easily shown as it was for directionless tracking in Section 4.

### 5.1 Terminal Sliding

The controls developed in (4) will ensure we remain in tracking mode despite the fact that $\beta_r$ and $\beta_\theta$ may be time-varying. However we now need to investigate the matter of whether the sliding is actually terminal. That is, does the robot slide to its target in finite time?

Under the conditions outlined above, a positive value for $\beta_r$ will cause sliding towards the target (consider $S_1 = \dot{r} + \beta_r r^{3/5} = 0$). A negative value will cause motion away from the target - the direct consequence of having an obstacle interposed between the robot and target necessitating a movement away from its target in the radial direction (this can be seen by considering the direction of steepest descent for $L$ under various conditions). The physical effects of $\beta_\theta$ in the presence of obstacles can be reasoned similarly.

To show that we still have finite time convergence to the target along the sliding mode, we first note that the robot is already in tracking mode. Thus $S_1 = S_2 = 0$ and

$$\dot{r} = -\beta_r r^{m/n} \qquad \dot{\theta} = \beta_\theta. \tag{16}$$

The velocity of the robot in the same relative polar coordinate frame is thus

$$\begin{aligned} \underset{\sim}{v} &= \dot{r}\underset{\sim}{e}_r + r\dot{\theta}\underset{\sim}{e}_\theta, \\ &= -(\beta_r r^{m/n})\underset{\sim}{e}_r + (r\beta_\theta)\underset{\sim}{e}_\theta. \\ &= (l_r r^{m/n})\underset{\sim}{e}_r + (l_\theta r)\underset{\sim}{e}_\theta. \end{aligned} \tag{17}$$

From the velocity equation, it can be observed that the robot does not precisely track the path of steepest descent, merely travels in a close approximation of its direction. It can be shown that choosing the $\beta_i$ so that the robot tracks the path of steepest descent precisely relaxing the finite time time convergence condition to an asymptotic (exponential) convergence.

Most importantly however, choosing the $\beta_i$ as in (15) does provide tracking of a path for which $L$ is strictly decreasing.

To see this, note that $L > 0$ at all legitimate points within the boundaries of the system, except at the target where $L = 0$. $L$ is also a strictly decreasing function (with respect to time) since

$$\begin{aligned} \dot{L} &= \underset{\sim}{\nabla} L \cdot \underset{\sim}{v} = -|\underset{\sim}{\nabla} L| \hat{\underset{\sim}{l}} \cdot \underset{\sim}{v} \\ &= -|\underset{\sim}{\nabla} L| \begin{pmatrix} l_r \\ l_\theta \end{pmatrix} \cdot \begin{pmatrix} l_r r^{m/n} \\ l_\theta r \end{pmatrix}, \\ &= -|\underset{\sim}{\nabla} L| (l_r^2 r^{m/n} + l_\theta^2 r) < 0, \end{aligned}$$

except at the target where $\dot{L} = 0$. Thus we have convergence to the target. Convergence in finite time can be verified by considering the following argument.

Since $\dot{L} < 0$ everywhere except at the target, the robot will approach and enter arbitrarily small neighbourhood (Liapunov asymptotic stability)of the target in finite time.

By construction $L$ has a local minimum at the target and thus there exists a small enough neighbourhood of the target such that the radial component of $\underset{\sim}{\nabla} L$ will be greater than zero at all points except at the local minima itself. Once the robot has entered this neighbourhood, $l_r < 0$ and subsequently $\beta_r > 0$ at all points within the neighbourhood.

To conclude, we may apply the same reasoning used in Section 4 to prove convergence occurs in finite time.

## 6 Robust Obstacle Avoidance

Prior to the commencement of tracking, directing the robot is not achieved in a precise manner and care must be taken to initialise the robot in a starting position with a velocity that will not cause it to career into an obstacle before it can begin tracking.

This is not unrealistic - placing a human in a vehicle with a high initial velocity directed towards a nearby wall will inevitably bring about an imminent crash given realistic conditions.

In this environment these conditions are simulated by the parameter $K$ in the control equations (4). Tracking mode can be achieved as quickly as desired by choosing a high enough value of $K$.

However the step sized used in a simulation, or a realistic bound on a control in a practical environment will limit the bounds we can place on $K$, leaving an opening where the robot will be in danger of colliding with an obstacle due to poor initialisation.

Another approach to achieving some early measure of obstacle avoidance is to scale the $\beta$ parameters as they approach an obstacle (note that here $\beta_r$ and $\beta_\theta$ are restricted to values between $-1$ and 1). This approach was used for a simpler problem in [4] and proved effective in avoiding external boundaries immediately after initialisation. Determining its robustness for obstacle avoidance with mobile targets as presented here is however, beyond the scope of this paper.

## 7 Examples

All of the examples to follow are simulated within an environment subject to the following constraints:

- *Two Robots* are controlled simultaneously to their respective targets.
- *Robots* trace out their individual paths (black).
- *Targets* are mobile and trace out circular paths (blue).
- *Boundaries* are placed at the frame of of a square window of width 6 units.
- *Obstacle (Stationary)* is placed at the origin (crow).
- *Obstacle (Mobile)* is represented by the opposing robot.
- *Obstacles* are all represented mathematically as a disc with pre-specified radius.
- *Penalty Multipliers* are all set to a value of $\lambda_i = 0.1$.
- *Sliding Constant* in each example was set to $K = 1.0$.

## 7.1 Boundary Avoidance

Since we do not have proper directional control before tracking mode is achieved (Section 6) it was important to test the robustness of the boundary avoidance near initialisation. The simulation illustrated in Figure 2 was run with the following initialisations.

| | | |
|---|---|---|
| Robot 1 (position) | = | (5.0,-3.5) |
| Robot 1 (velocity) | = | (0.4,-1.8) |
| Robot 2 (position) | = | (-5.0,3.5) |
| Robot 2 (velocity) | = | (-1.2,0.0) |

Acceleral controls were minimal ($< 6$ units), smooth and quickly diminished once tracking mode was achieved (within 2 seconds). Convergence occured in approximately 5 seconds. These traits are greatly improved compared to [4, 5] and are common to all the examples presented here.

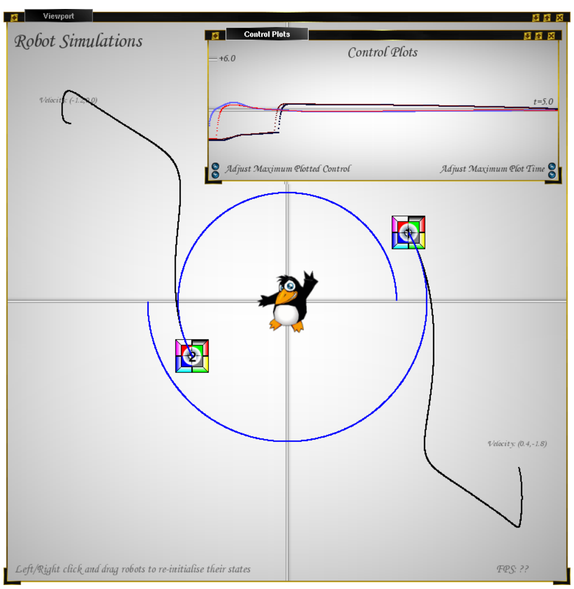


Figure 2: Boundary Avoidance

Increasing $K$ in similar simulations improved convergence response to the tracking mode as expected, thus allowing for a greater range of initial velocities. The responsiveness of varying the parameter $K$ however was observed not to be as immediately effective in initial boundary avoidance as a scaling of the $\beta_i$. More robust assurances would be gained with a more detailed analysis of the concepts presented in Secion 6.

## 7.2 Avoidance of a Stationary Obstacle

Avoidance of the stationary obstacle (the crow at the origin) is illustrated in Figure 3. Both robots divert from a direct approach as the stationary obstacle becomes a prominent feature between the robot and its target. Accelerational control was once again minimal ($< 4$) and convergence to the target occured within 4 seconds.

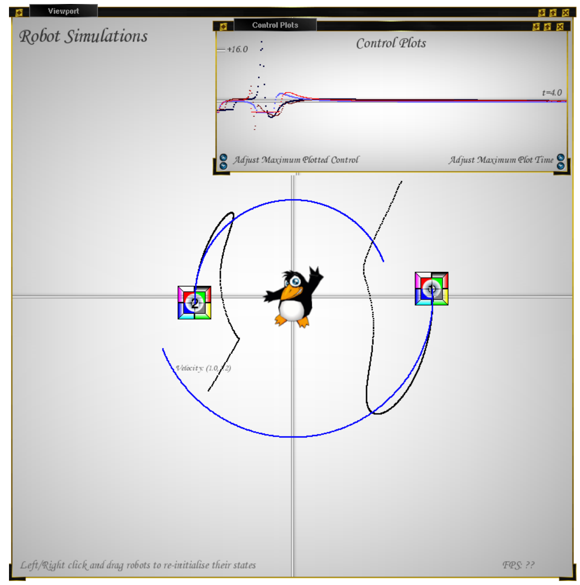


Figure 3: Obstacle Avoidance

## 7.3 Avoidance of a Mobile Obstacle

Targets (cross-hairs) move in a circular fashion around the stationary obstacle at the origin. Robot 1 is initialised close to its target to ensure prompt convergence occurs. Robot 2 is initialised so that the Robot 1 (mobile obstacle) and the stationary obstacle interpose themselves between Robot 2 and its target at critical points in its path. Progress snapshots for this simulation are illustrated in Figures 4-6.

2

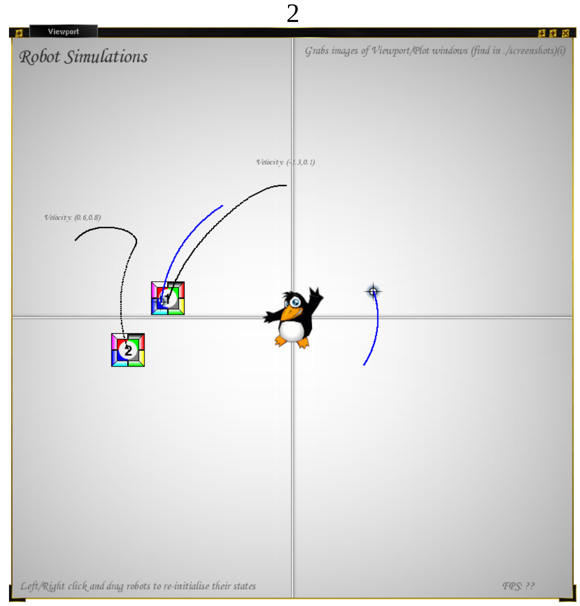


Figure 4: Accelerating Around the Mobile Obstacle

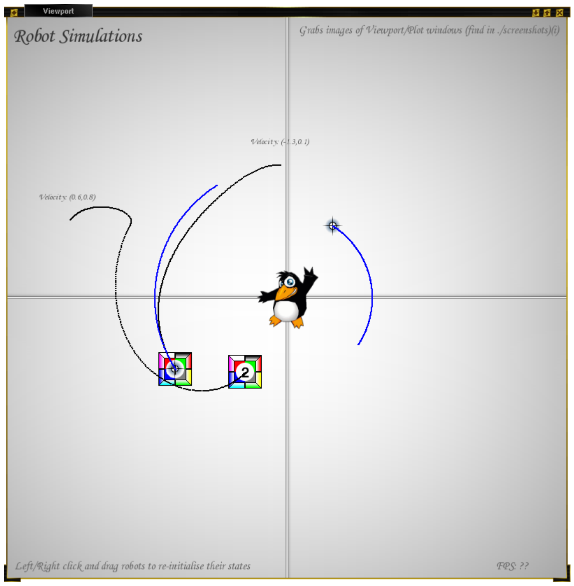


Figure 5: Confronted by the Stationary Obstacle

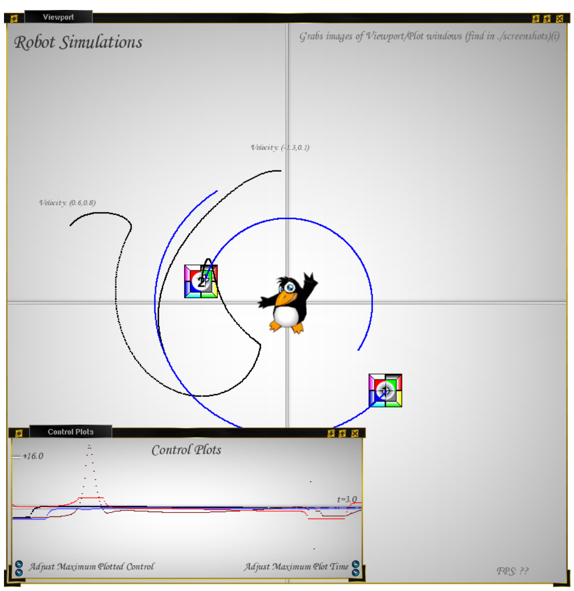


Figure 6: Final Convergence to the Target

Here the second robot undergoes a period of angular acceleration (peaking at a magnitude of 17 - this is the spike that can be seen in the control plots) in order to move around and avoid the first robot. Convergence occured in less than 4 seconds.

### 7.4 Examples - Conclusions

In addition to being free from the initialisation discontinities present in [4, 5], all the examples presented here have much improved convergence times (less than half) when compared with the algorithms used in [4, 5] possessing similar parameter constraints on $K$ and the $\beta_i$. This is primarily due to the directional tracking providing much more 'intelligent paths' for the robots to follow that involve a less circuitous route (previously the control would drive the system variables to convergence, but each would be done so independently and consequently a less optimal path was generally chosen).

Accelerational control in each example is relatively smooth and minimal. Once tracking mode is achieved, the control magnitudes are negligible - the switch to tracking mode can easily be determined from the output plots where the controls drop sharply and settle down to relatively minor values which only vary as mobile obstacles interfere or the target disappears behind an obstacle.

## 8 Conclusion

The method proposed for control of a simulated point mass robot with acceleration control, allows us to achieve directional control whilst maintaining finite time convergence to the target. It does this free from the initialisation discontinuities associated with the method in [5] and also provides an algorithm with formal proof for the finite time convergence along sliding and angular tracking modes.

## References

[1] R.J. Stonier. Use of liapunov techniques for collision-avoidance of robot arms. In *Advances in Control and Dynamic Systems, Vol. 34, No. 2*, pages 185–215, 1990.

[2] X. Yu Y. Feng and Z. Man. Non-singular terminal sliding mode control of rigid manipulators. In *Automatica, Vol. 38(12)*, pages 2159–2167, 2002.

[3] X. Yu and Z. Man. Variable structure systems with terminal sliding modes. In *Lecture Notes in Control and Information Sciences, Vol. 274*, pages 109–128. Springer-Verlag, 2002.

[4] D.J. Stonier and R.J. Stonier. Handling constraint avoidance and finite time switching control for simulated mobile robots. In *Proceedings of the 2nd International Conference on Robotics and Autonomous Systems*. CIRAS, Singapore, on CD ISSN:0219-613, PS02-4-01, 2003.

[5] D.J. Stonier and R.J. Stonier. c̈onstraint avoidance and finite time control for simulated robots using a gradient calculation.̈ In *Proceedings of the PICS International Conference and Exhibition on Instrumentation and Control*. PICS-ICEIC, Manilla, 2003.