Robotica (2016) volume 34, pp. 1071–1089.
doi:10.1017/S0263574714002070

A new stabilizing solution for motion planning and control of multiple robots

Avinesh Prasad, Bibhya Sharma* and Jito Vanualailai

School of Computing, Information & Mathematical Sciences, University of the South Pacific, Suva, FIJI

(Accepted July 7, 2014. First published online: August 19, 2014)

SUMMARY

This paper formulates a new scalable algorithm for motion planning and control of multiple point-mass robots. These autonomous robots are designated to move safely to their goals in *a priori* known workspace cluttered with fixed and moving obstacles of arbitrary positions and sizes. The control laws proposed for obstacle and collision avoidance and target convergence ensure that the equilibrium point of the given system is asymptotically stable. Computer simulations with the proposed technique and applications to a team of two planar (RP) manipulators working together in a common workspace are presented. Also, the robustness of the system in the presence of noise is verified through simulations.

KEYWORDS: Point-mass robots; Motion planning; Planar robot arms; Asymptotic stability.

1. Introduction

Active and continuous research in the area of motion planning and control (MPC) of robots has incessantly spanned over the past two decades. A high level of sustained interest in this field is invariably due to the coupling of inherent constraints and restrictions, wide-ranging capabilities of robots, abundance of real-world applications[1,2] and the array of possibilities of mechanical systems. Nowadays, robots are capable of performing dull, dirty, dangerous or difficult tasks such as surveillance, construction, transportation and traffic control, healthcare, mining and sampling, reconnaissance, landscape maintenance, museum guides and planetary exploration.[1,3–5] Developing multiple robots are favored over single units due to the stringent time and cost constraints, increased robustness, greater fault tolerance, better safety, accelerated performances and higher capabilities, to outline a few major ones.[2,3,6] The multiple robots can cooperate and network for better, faster and more efficient results.

The aforementioned tasks and missions of multi-robots are normally carried out in dynamic environments which includes both stationary and dynamic obstacles. The dynamic obstacles (known or even unknown) may be the mobile robots themselves as well as other solid bodies moving in the workspace. Constructing control laws is a difficult and challenging task because the environment is dynamic rather than static.[1] Over the past two decades, researchers have devised numerous algorithms to address motion planning problem for multiple robots taking into account both collision and obstacle avoidances. These algorithms have been categorized into continuous, piecewise continuous, discontinuous, discrete, time varying and a working hybrid of controllers.[1,7] The reader can refer to[1] for more information on these types of algorithms.

Most of the work carried out with autonomous multiple agents in obstacle ridden workspace using continuous control laws in dynamic environments have guaranteed stability only. This implies that there is a possibility that some trajectories starting in the neighborhood of the equilibrium point may lead to traps (local minima) outside the equilibrium point. Thus one desires to have trajectories that always converge to the equilibrium, that is to attain an asymptotic stable system. Researchers over the years have proposed some useful techniques to solve the problem of local minima via the use of some

* Corresponding author. E-mail: sharma_b@usp.ac.fj

special functions.[1] These techniques are using Potential Functions,[8] using Dipolar Inverse Lyapunov Functions,[9] executing a random robot motion,[10] temporarily relocating the goal[11] and constructing a potential field based on superquadrics.[12]

An interesting and noteworthy work was done by Vanualailai *et al.*[13] in 2008 where the authors showed that if the robot, its target and the obstacle positions are collinear, then the robot can be *trapped* behind the obstacle. This implies that the asymptotic stability of a system is dependent on the initial conditions. Thus by removing those initial conditions which can lead to traps, Vanualailai and his colleagues successfully proved the asymptotic stabilization of a point mass system.[13]

In this paper, we will control the motion of multiple point-masses in the presence of fixed and moving obstacles in *a priori* known workspace. Our seminal aim is to design continuous control laws that ensure asymptotic stability of the system, irrespective of the initial conditions of the system (as opposed to[13]). A scalable algorithm for obstacle and collision avoidances, and target convergence is proposed that works for multiple point-mass robots with multiple moving and fixed obstacles of arbitrary shapes and sizes. To the authors' knowledge, this is the first time such an algorithm is developed for avoidances of fixed and moving obstacles in parallel. In addition, we also look at the effect of noise in the simulations. These noise are time-dependent small disturbances that could be encountered in the sensor readings.

A new systematic control scheme is described for the construction of the control laws which can be favored over other schemes. The method is systematic, elegant and yet simple compared to, for example, the Lyapunov-based control scheme[1,2,13] where there is no definite and standard procedure of constructing a Lyapunov function from which the controllers can be extracted. Moreover, the scalable algorithm proposed in this paper for multiple point-masses can easily be applied to other planar robots such as planar robot arms, car-like robots and mobile manipulators. As an illustration, we have considered the motion of two planar (RP) robot working together in Section 6.

This paper is organized as follows. In Section 2, we define of the workspace, point-mass robots and derive the kinematic model. The motion planning and control problem of the point masses is discussed in Section 3, together with introduction of the targets and the velocity algorithm. In Section 4, various types of obstacles are considered and the obstacle avoidance scheme is proposed. Stability analysis is carried out in Section 5 while Section 6 considered the motion of two planar (RP) robot working together in a common workspace. Finally, in Section 7 concluding remarks on the contributions and future work are made.

2. Modelling a Point-Mass Robot

The notations and terminologies of this paper are adopted from the prequel.[14]

Definition 1. *The workspace is a fixed, closed and bounded rectangular region for some* $\eta_1 > 0$ *and* $\eta_2 > 0$. *Precisely, the workspace is the set* $WS = \{(z_1, z_2) \in \mathbb{R}^2 : 0 \leq z_1 \leq \eta_1, 0 \leq z_2 \leq \eta_2\}$.

Definition 2. *Let* P_i *be the* i*th* point-mass robot *in the* z_1z_2 *plane, positioned at* (x_i, y_i) *with a circular protective region of radius* $r_{P_i} \geq 0$ *and moving with a velocity of* v_i *at time* $t \geq 0$. *Precisely, the circular protective region is the set*

$$P_i = \left\{(z_1, z_2) \in \mathbb{R}^2 : (z_1 - x_i)^2 + (z_2 - y_i)^2 \leq r_{P_i}^2\right\}.$$

According to,[1] the disk-representation strategically aids in the construction of the motion planning algorithms. Let u_{i1} and u_{i2} be the z_1 and z_2 components, respectively, of v_i, then the kinematic model of P_i can be expressed as

$$\left.\begin{aligned} \dot{x}_i &= u_{i1}, \\ \dot{y}_i &= u_{i2}, \\ (x_{i_0}, y_{i_0}) &:= (x_i(0), y_i(0)) \end{aligned}\right\} \tag{1}$$

https://doi.org/10.1017/S0263574714002070

for $i = 1, 2, \ldots, n$. System (1) is a description of the instantaneous velocities of P_i where u_{i1} and u_{i2} are classified as the controllers. Hereafter, we shall use the vector notation $\mathbf{x}_i = (x_i(t), y_i(t))$ to refer to the position of P_i in the z_1z_2 plane.

3. Convergence of Point Masses

In our MPC, we want the ith point-mass robot P_i to start from an initial position, move towards a target and finally converge at the center of the target. We therefore affix a target for each P_i:

Definition 3. *The target for the point-mass robot P_i is a disk of center (p_{i1}, p_{i2}) and radius r_{T_i}. Precisely, it is a set*

$$T_i = \left\{(z_1, z_2) \in \mathbb{R}^2 : (z_1 - p_{i1})^2 + (z_2 - p_{i2})^2 \leq r_{T_i}^2\right\}.$$

We consider an appropriate form of v_i, which can drive P_i from its initial position to the target position and make it stop there. The authors in ref. [14] developed a practical velocity algorithm which depended on the initial and final positions of the robot:

$$v_i(t) = |v_0| \frac{\|\mathbf{x}_i(t) - \mathbf{e}_i\|}{\|\mathbf{x}_i(0) - \mathbf{e}_i\|}, \tag{2}$$

where v_0 is the initial velocity (assumed same for all the robots) of P_i at $t = 0$ and $\mathbf{e}_i = (p_{i1}, p_{i2}) \neq \mathbf{x}_i(0)$ is an equilibrium point of system (1). Note that v_i given by equation (2) is defined, continuous and positive over the domain

$$D_i = \{\mathbf{x} \in \mathbb{R}^2 : \ \mathbf{x}_i(0) \neq \mathbf{e}_i\}.$$

For $\mathbf{x}_i(t) \neq \mathbf{e}_i$, we further let $\xi_i(t)$ be the angular position of T_i with respect to the position of P_i at time t. The angle $\xi_i(t)$ is defined implicitly as

$$\tan \xi_i(t) = \frac{p_{i2} - y_i(t)}{p_{i1} - x_i(t)}. \tag{3}$$

4. Collision and Obstacle Avoidances

The various types of fixed and moving obstacles and new stabilizing controllers for the avoidances are outline in this section. The obstacles considered in this paper include

- moving obstacles, which are the point-mass robots;
- fixed obstacles, which are antitargets and stationary solid objects of various shapes and sizes.

This paper describes all possible obstacles using simple forms such as circles, ellipses and lines, which can enclose all the possible obstacles.

Assumption 1. *Initially, at $t = 0$, there is no positional overlap between any two robots or between a robot and an obstacle in the workspace WS.*

Assumption 2. *There is a priori knowledge of the workspace. That is, the initial and target positions of the robots, and the types, positions and sizes of all obstacles are priori known.*

4.1. Moving obstacles

From a practical viewpoint, avoidance of moving obstacles is the most important task for the mobile robots[1] operating in a dynamic environment. A mobile robot itself becomes a moving obstacle for all the other mobile robots in WS. We provide the following definition of a moving obstacle.

Definition 4. *The jth moving obstacle is a disk with center (x_j, y_j) and radius r_{Pj}. Precisely, the jth moving obstacle is the set $P_j = \{(z_1, z_2) \in \mathbb{R}^2 : (z_1 - x_j)^2 + (z_2 - y_j)^2 \leq r_{Pj}^2\}$.*

https://doi.org/10.1017/S0263574714002070

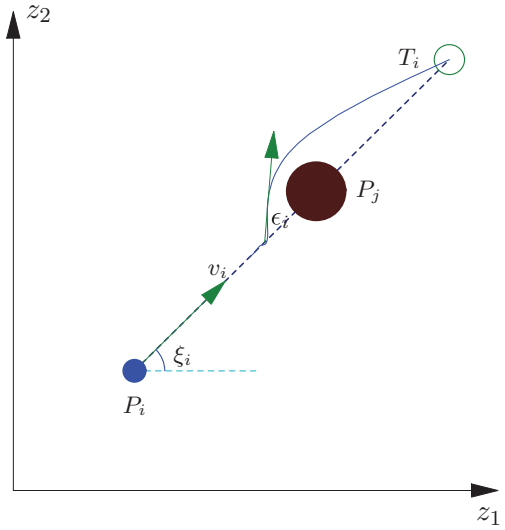


Fig. 1. Schematic representation of the collision avoidance of P_j by P_i in the z_1z_2 plane.

In order for the point-mass robot P_i to avoid collision with P_j, we define controllers u_{i1} and u_{i2} as:

$$\begin{cases} u_{i1} = v_i \cos(\xi_i + \epsilon_i), \\ u_{i2} = v_i \sin(\xi_i + \epsilon_i), \end{cases} \tag{4}$$

where ϵ_i determines the direction in which P_i should turn to avoid collision with P_j. The inclusion of ϵ_i is well elucidated in Fig. 1. If $\epsilon_i > 0$, then the point-mass will turn left; if $\epsilon_i < 0$, it will turn right; and if $\epsilon_i = 0$, it will move straight towards the target. Thus controlling the value of ϵ_i will enable robot P_i to avoid obstacles and reach its target safely.

Claim 1 *With the form of the velocities given by equation (4), the point-mass robot P_i is guaranteed to converge at its target position.*

Proof. When the ith robot reaches its target, (p_{i1}, p_{i2}), the controllers u_{i1} and u_{i2} in equation (4) converge to zero since v_i given in (2) vanishes at the target.

Definition 5. *Let $d_{\max} > 0$ be a predefined scalar. The set $\mathscr{S}_1$ defined by*

$$\mathscr{S}_1 = \bigcup_{\substack{j=1 \\ j\neq i}}^{n} \left\{ (z_1, z_2) \in \mathbb{R}^2 : r_{p_j}^2 < (z_1 - x_j)^2 + (z_2 - y_j)^2 < (r_{p_j} + d_{\max})^2 \right\}$$

is denoted as the total sensing zone which is the combination of all sensing zones as seen in Fig. 2.

Let

$$R_{ij}^{(1)} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} - (r_{p_i} + r_{p_j})$$

be the distance between P_i and P_j for $j = 1, 2, \ldots, n,\ j \neq i$. Consider $R_{i1} = \min(R_{ij}^{(1)})$ for $j = 1, 2, \ldots, n,\ j \neq i$, the distance from P_i to the nearest P_j. For this nearest moving obstacle, denote its center as (x', y') and define

$$\tan \varphi_{i1} = \frac{y' - y_i}{x' - x_i} \quad \text{and}$$
$$f_{i1} = (x_i - x')(p_{i2} - y_i) - (y_i - y')(p_{i1} - x_i).$$

https://doi.org/10.1017/S0263574714002070

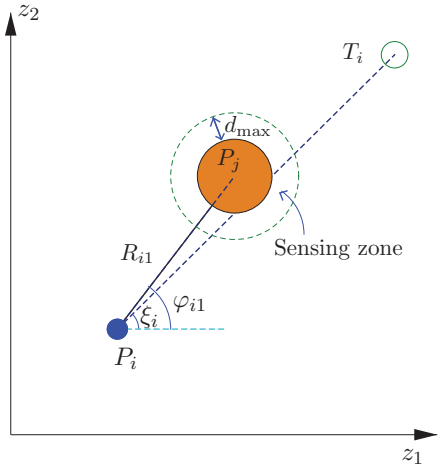


Fig. 2. Schematic representation of the avoidance scheme with parameter $d_{\max}$.

Remark: If $f_{i1} = 0$, then the points (x_i, y_i), (x', y') and (p_{i1}, p_{i2}) are collinear, so the angle φ_{i1} shown in Fig. 2 will be same as ξ_i. If $f_{i1} < 0$, then $\varphi_{i1} > \xi_i$ and if $f_{i1} > 0$, then $\varphi_{i1} < \xi_i$.

We now look for an appropriate form of ϵ_i. With the help of Fig. 2, we enact the following rules:

Rule 1: In order to avoid collision with the nearest moving obstacle, R_{i1} should be positive.
Rule 2: If the robot is approaching the obstacle, it should change its direction when it enters the sensing zone.
Rule 3: When the robot enters the sensing zone, it should turn right if $f_{i1} \leq 0$. Otherwise it should turn left. This is to ensure that it follows the shortest path.

Note that the size of the sensing zone is determined by $d_{\max}$. If $d_{\max}$ is large, then P_i will avoid the moving obstacle from a greater distance. Thus $d_{\max}$ is classified as the *control* parameter in this research.

4.1.1. Proposed form of ϵ_i. Normally seen in literature[1,2,13,15] for effective avoidances, the distance R_{i1} and similar avoidance measures appear in the denominator of repulsive potential field functions. Adopting the methodology, we propose the following form of ϵ_i:

$$\epsilon_i = \tan^{-1}\left(\frac{\alpha_{i1}\beta_{i1}}{R_{i1}}\right), \tag{5}$$

where

$$\alpha_{i1} = \begin{cases} 0, & \text{if } R_{i1} \geq d_{\max} \\ d_{\max} - R_{i1}, & \text{if } R_{i1} < d_{\max} \end{cases} \quad \text{and} \quad \beta_{i1} = \begin{cases} 1, \text{ if } f_{i1} \leq 0 \\ -1, \text{ if } f_{i1} > 0 \end{cases}.$$

The function α_{i1} plays two roles here. Firstly, it ensures that the output ϵ_i will be a continuous function. Thus the controllers derived will be continuous everywhere along the trajectory of the system. Secondly, it ensures that the turning is initiated when a robot enters the sensing zone. The parameter β_{i1} is an *indicator function*, which indicates the direction P_i should turn in the sensing zone to ensure that an overall shortest path is achieved. We also note that:

1. The function ϵ_i given by equation (5) spans in the interval $(-\pi/2, \pi/2)$.
2. With the form of ϵ_i given in equation (5), we see that as P_i comes closer to P_j, the R_{i1} will decrease. This will inevitably increase $|\epsilon_i|$ since R_{i1} appears in the denominator. Hence an increase in $|\epsilon_i|$ will force P_i will move away from the obstacle.

https://doi.org/10.1017/S0263574714002070

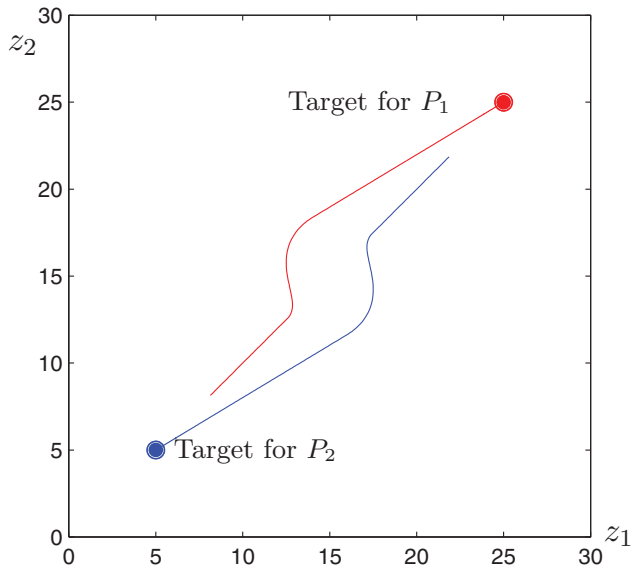


Fig. 3. Trajectory of two point-mass robots with $d_{\max} = 3$ and $v_0 = 5$.

Now, substituting (5) into (4) and simplifying, we obtain controllers of the form

$$\begin{cases} u_{i1} = \dfrac{|v_0|}{\|\mathbf{x}_i(0) - \mathbf{e}_i\|} \left[\dfrac{(p_{i1} - x_i)R_{i1} - (p_{i2} - y_i)\alpha_{i1}\beta_{i1}}{\sqrt{\alpha_{i1}^2 + R_{i1}^2}} \right] \\ u_{i2} = \dfrac{|v_0|}{\|\mathbf{x}_i(0) - \mathbf{e}_i\|} \left[\dfrac{(p_{i2} - y_i)R_{i1} + (p_{i1} - x_i)\alpha_{i1}\beta_{i1}}{\sqrt{\alpha_{i1}^2 + R_{i1}^2}} \right] \end{cases} \tag{6}$$

We further note that these controllers are bounded and continuous at every point over the domain

$$D_{i1} = \{(x_i, y_i) \in \mathbb{R}^2 : \mathbf{x}_i(0) \neq \mathbf{e}_i \cap (x_i - x_j)^2 + (y_i - y_j)^2 > (r_{p_i} + r_{p_j})^2 \text{ for } j = 1, 2, \ldots, n\}.$$

Example 1. *We demonstrate the simulation result for two point mass mobile robots navigating in the z_1z_2 plane. Figure 3 shows the trajectories of these robots from the initial to final states. The initial and final states are given as: $\mathbf{x}_1(0) = (8, 8)$, $\mathbf{x}_2(0) = (22, 22)$ and $\mathbf{e}_1 = (25, 25)$, $\mathbf{e}_2 = (5, 5)$, respectively When the robots approach each other in the middle of the workspace, we clearly see that they avoid each other, and finally converge to their designated targets.*

4.2. Antitargets: Targets as obstacles

According to,[1] a target fixed for a mobile robot needs to be treated as a stationary obstacle for all the remaining robots operating within WS. Therefore, the target of P_i will inherently become an antitarget for all P_j's, for $j = 1, 2, \ldots, n$, $j \neq i$ in WS.

Definition 6. *The jth antitarget is a disk with center (p_{j1}, p_{j2}) and radius r_{Tj}. It is described as the set $AT_j = \{(z_1, z_2) \in \mathbb{R}^2 : (z_1 - p_{j1})^2 + (z_2 - p_{j2})^2 \leq r_{Tj}^2\}$.*

We can then describe the sensing zone surrounding the antitargets as

$$\mathscr{S}_2 = \bigcup_{\substack{j=1 \\ j \neq i}}^{n} \left\{ (z_1, z_2) \in \mathbb{R}^2 : r_{Tj}^2 < (z_1 - p_{j1})^2 + (z_2 - p_{j2})^2 < (r_{Tj} + d_{\max})^2 \right\}.$$

https://doi.org/10.1017/S0263574714002070

Let $R_{ij}^{(2)} = \sqrt{(x_i - p_{j1})^2 + (y_i - p_{j2})^2} - (r_{p_i} + r_{T_j})$ be the distance between P_i and AT_j. Now, consider $R_{i2} = \min(R_{ij}^{(2)})$ for $j = 1, 2, \ldots, n,\ j \neq i$ which is the distance from P_i to the *nearest* jth antitarget. For this nearest antitarget at every $t \geq 0$, denote its center as (a', b') and let

$$f_{i2} = (x_i - a')(p_{i2} - y_i) - (y_i - b')(p_{i1} - x_i).$$

Then define ϵ_i as:

$$\epsilon_i = \tan^{-1}\left(\frac{\alpha_{i2}\beta_{i2}}{R_{i2}}\right),$$

where

$$\alpha_{i2} = \begin{cases} 0, & \text{if } R_{i2} \geq d_{\max} \\ d_{\max} - R_{i2}, & \text{if } R_{i2} < d_{\max} \end{cases} \quad \text{and} \quad \beta_{i2} = \begin{cases} 1, \text{ if } f_{i2} \leq 0 \\ -1, \text{ if } f_{i2} > 0 \end{cases}.$$

Thus for the avoidance of the antitargets, the controllers are

$$\begin{cases} u_{i1} = \dfrac{|v_0|}{\|\mathbf{x}_i(0) - \mathbf{e}_i\|}\left[\dfrac{(p_{i1} - x_i)R_{i2} - (p_{i2} - y_i)\alpha_{i2}\beta_{i2}}{\sqrt{\alpha_{i2}^2 + R_{i2}^2}}\right] \\ u_{i2} = \dfrac{|v_0|}{\|\mathbf{x}_i(0) - \mathbf{e}_i\|}\left[\dfrac{(p_{i2} - y_i)R_{i2} + (p_{i1} - x_i)\alpha_{i2}\beta_{i2}}{\sqrt{\alpha_{i2}^2 + R_{i2}^2}}\right] \end{cases} \tag{7}$$

which are bounded and continuous at every point over the domain

$$D_{i2} = \{(x_i, y_i) \in \mathbb{R}^2 : \mathbf{x}_i(0) \neq \mathbf{e}_i \cap (x_i - p_{j1})^2 + (y_i - p_{j2})^2 > (r_{p_i} + r_{T_j})^2 \text{ for } j = 1, 2, \ldots, n\}.$$

Example 2. *Figure 4 shows an interesting simulation with four point-mass robots. We see that each of the robots maneuver from its initial position to the target position while avoiding the targets of the other robots that may be encountered along its path. The initial and final positions were generated randomly. Figure 5 shows evolution of the controllers for P_1. One can clearly notice the asymptotic convergence of the controllers as $t \to \infty$. For the other robots a similar trend in evolution was obtained.*

4.3. Stationary elliptical obstacles

Let $q \in \mathbb{N}$ solid bodies be fixed within the boundaries of WS which may intersect the path of the point-mass robots. We assume that the lth obstacle is an ellipse with center given as (o_{l1}, o_{l2}).

Definition 7. *The lth obstacle is a set*

$$FO_l = \left\{(z_1, z_2) \in \mathbb{R}^2 : \frac{(z_1 - o_{l1})^2}{a_l^2} + \frac{(z_2 - o_{l2})^2}{b_l^2} \leq 1\right\}, \quad \textit{for } l = 1, 2, \ldots, q,$$

where a_l and b_l are positive constant.

In special case if $a_l = b_l$, then the set FO_l represents a circular object. The sensing zone surrounding these elliptical obstacles is

$$\mathcal{S}_3 = \bigcup_{l=1}^{q}\left\{(z_1, z_2) \in \mathbb{R}^2 : \frac{(z_1 - o_{l1})^2}{(a_l + d_{\max})^2} + (\frac{z_2 - o_{l2})^2}{(b_l + d_{\max})^2} < 1\right\}.$$

https://doi.org/10.1017/S0263574714002070

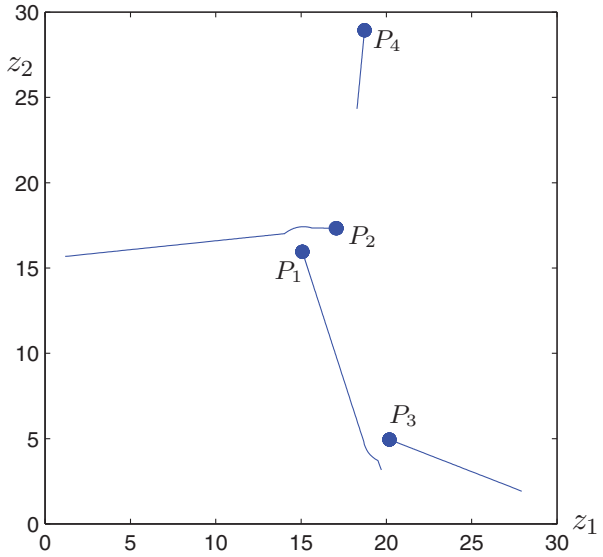


Fig. 4. Trajectory of four point-mass robots with $d_{\max} = 3$ and $v_0 = 5$.

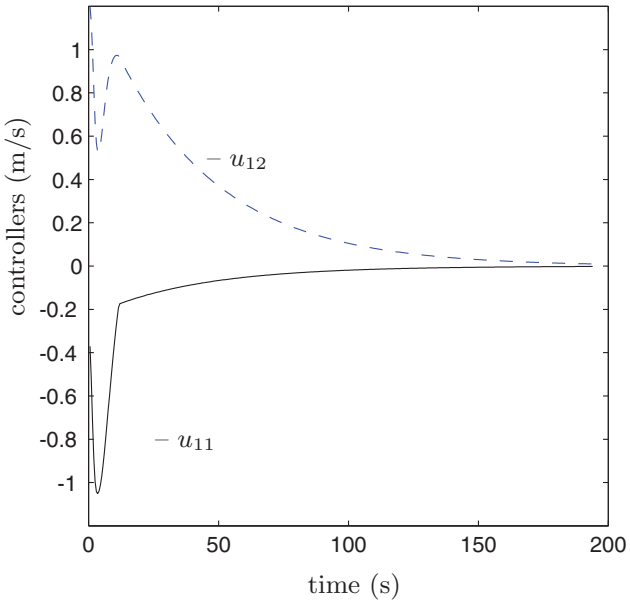


Fig. 5. Evolution of the control signals, u_{11} and u_{12} along the trajectory of P_1.

In order to avoid all the obstacles in the workspace, it is sufficient for the robots to avoid the obstacle that is nearest to it at any time $t \geq 0$. As such, we let

$$R_{il}^{(3)} = \frac{(x_i - o_{l1})^2}{(a_l + r_{p_i})^2} + \frac{(y_i - o_{l2})^2}{(b_l + r_{p_i})^2} - 1$$

be a distance measure between P_i and FO_l and consider $R_{i3} = \min(R_{i1}^{(3)}, R_{i2}^{(3)}, \ldots, R_{iq}^{(3)})$ being the distance measure from P_i to the *nearest* stationary elliptical obstacle. For this nearest obstacle, denote its center as (o_1', o_2') and let $f_{i3} = (x_i - o_1')(p_{i2} - y_i) - (y_i - o_2')(p_{i1} - x_i)$. Further let $\alpha_{i3} =$

https://doi.org/10.1017/S0263574714002070

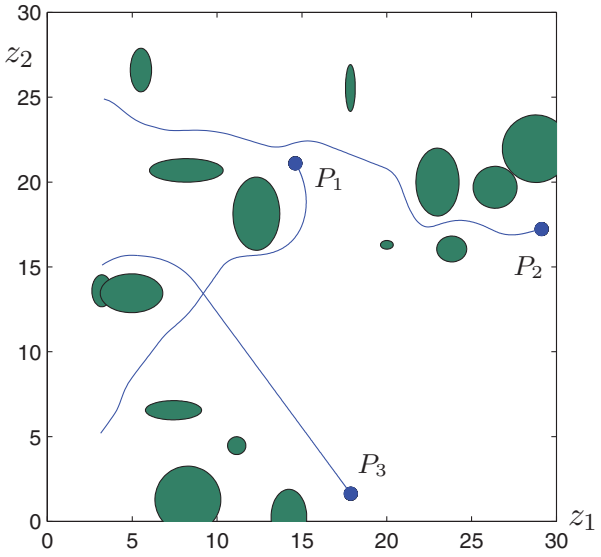


Fig. 6. Collision-free trajectories of three point-mass robots in an obstacle-ridden workspace with $d_{\max} = 3$ and $v_0 = 5$.

$\begin{cases} 0, & \text{if } R_{i3} \geq d_{\max} \\ d_{\max} - R_{i3}, & \text{if } R_{i3} < d_{\max} \end{cases}$ and $\beta_{i3} = \begin{cases} 1, & \text{if } f_{i3} \leq 0 \\ -1, & \text{if } f_{i3} > 0 \end{cases}$. In this case, we define ϵ_i as:

$$\epsilon_i = \tan^{-1}\left(\frac{\alpha_{i3}\beta_{i3}}{R_{i3}}\right)$$

so that the controllers u_{i1} and u_{i2} become

$$\begin{cases} u_{i1} = \dfrac{|v_0|}{\|\mathbf{x}_i(0) - \mathbf{e}_i\|}\left[\dfrac{(p_{i1} - x_i)R_{i3} - (p_{i2} - y_i)\alpha_{i3}\beta_{i3}}{\sqrt{\alpha_{i3}^2 + R_{i3}^2}}\right] \\ u_{i2} = \dfrac{|v_0|}{\|\mathbf{x}_i(0) - \mathbf{e}_i\|}\left[\dfrac{(p_{i2} - y_i)R_{i3} + (p_{i1} - x_i)\alpha_{i3}\beta_{i3}}{\sqrt{\alpha_{i3}^2 + R_{i3}^2}}\right] \end{cases} \tag{8}$$

which are bounded and continuous at every point over the domain

$$D_{i3} = \{(x_i, y_i) \in \mathbb{R}^2 : \mathbf{x}_i(0) \neq \mathbf{e}_i \cap \frac{(x - o_{l1})^2}{(a_l + r_{p_i})^2} + \frac{(y - o_{l2})^2}{(b_l + r_{p_i})^2} > 1$$
$$\text{for } l = 1, 2, \ldots, q\}.$$

Example 3. *Figure 6 illustrates a simulation where multiple point-masses move in a workspace cluttered with fixed elliptical obstacles. The size and position of the obstacles are randomly generated. The robots avoid each other and the fixed elliptical obstacles they encounter in their path to the target.*

4.4. Line obstacles

A line segment can be considered as a fixed obstacle in Euclidian plane. Let us fix $m > 0$ line obstacles in WS.

https://doi.org/10.1017/S0263574714002070

Definition 8. *The kth line segment in the z_1z_2 plane, from the point (a_{k1}, b_{k1}) to the point (a_{k2}, b_{k2}) is the set*

$$LO_k = \{(z_1, z_2) \in \mathbb{R}^2 : (z_1 - X_k)^2 + (z_2 - Y_k)^2 = 0\}, \text{ for } k = 1, 2, \ldots, m,$$

where $X_k = a_{k1} + (a_{k2} - a_{k1})\lambda_k$ and $Y_k = b_{k1} + (b_{k2} - b_{k1})\lambda_k$ is its parametric representation for $0 \le \lambda_k \le 1$.

With the help of Definition 8, we can then describe the sensing zone that encloses the line segments as

$$\mathscr{S}_4 = \bigcup_{k=1}^{m} \left\{(z_1, z_2) \in \mathbb{R}^2 : 0 < (z_1 - X_k)^2 + (z_2 - Y_k)^2 < d_{\max}^2\right\}.$$

For the robot P_i to avoid the kth line segment, we utilize the minimum distance technique (MDT) designed by Sharma in ref. [1]. The technique basically involves calculating the minimum distance from a robot to a line segment is calculated and hence avoiding the resultant closest point. Avoidance of the closest point on a line segment simply affirms that the mobile robot avoids the whole line segment.

Minimizing the Euclidian distance between the point (x_i, y_i) of P_i and the point (X_k, Y_k) on the kth line segment, we get

$$\lambda_{ik} = (x_i - a_{k1})q_{k1} + (y_i - b_{k1})q_{k2},$$

where

$$q_{k1} = \frac{(a_{k2} - a_{k1})}{(a_{k2} - a_{k1})^2 + (b_{k2} - b_{k1})^2}, \quad \text{and} \quad q_{k2} = \frac{(b_{k2} - b_{k1})}{(a_{k2} - a_{k1})^2 + (b_{k2} - b_{k1})^2}.$$

If $\lambda_{ik} \ge 1$, then we let $\lambda_{ik} = 1$, in which case $(X_k, Y_k) = (a_{k2}, b_{k2})$ and if $\lambda_{ik} \le 0$, then we let $\lambda_{ik} = 0$, in which case $(X_k, Y_k) = (a_{k1}, b_{k1})$. Otherwise we accept the value of λ_k between 0 and 1.[1]

We let $R_{ik}^{(4)} = \sqrt{(x_i - X_k)^2 + (y_i - Y_k)^2} - r_{p_i}$ be the distance between P_i and LO_k and consider $R_{i4} = \min(R_{i1}^{(4)}, R_{i2}^{(4)}, \ldots, R_{i5}^{(4)})$ being the distance measure from P_i to the *nearest* stationary line obstacle. For this nearest obstacle, denote its center as (X', Y') and define

$$f_{i4} = (x_i - X')(p_{i2} - y_i) - (y_i - Y)(p_{i1} - x_i),$$

$$\alpha_{i4} = \begin{cases} 0, & \text{if } R_{i4} \ge d_{\max} \\ d_{\max} - R_{i4}, & \text{if } R_{i4} < d_{\max} \end{cases} \quad \text{and}$$

$$\beta_{i4} = \begin{cases} 1, & \text{if } f_{i4} \ge 0 \\ -1, & \text{if } f_{i4} < 0 \end{cases}.$$

For P_i to avoid the line segments, we define ϵ_i as:

$$\epsilon_i = \tan^{-1}\left(\frac{\alpha_{i4}\beta_{i4}}{R_{i4}}\right)$$

so that the controllers u_{i1} and u_{i2} are

$$\begin{cases} u_{i1} = \dfrac{|v_0|}{\|\mathbf{x}_i(0) - \mathbf{e}_i\|}\left[\dfrac{(p_{i1} - x_i)R_{i4} - (p_{i2} - y_i)\alpha_{i4}\beta_{i4}}{\sqrt{\alpha_{i4}^2 + R_{i4}^2}}\right] \\ u_{i2} = \dfrac{|v_0|}{\|\mathbf{x}_i(0) - \mathbf{e}_i\|}\left[\dfrac{(p_{i2} - y_i)R_{i4} + (p_{i1} - x_i)\alpha_{i4}\beta_{i4}}{\sqrt{\alpha_{i4}^2 + R_{i4}^2}}\right] \end{cases} \tag{9}$$

https://doi.org/10.1017/S0263574714002070

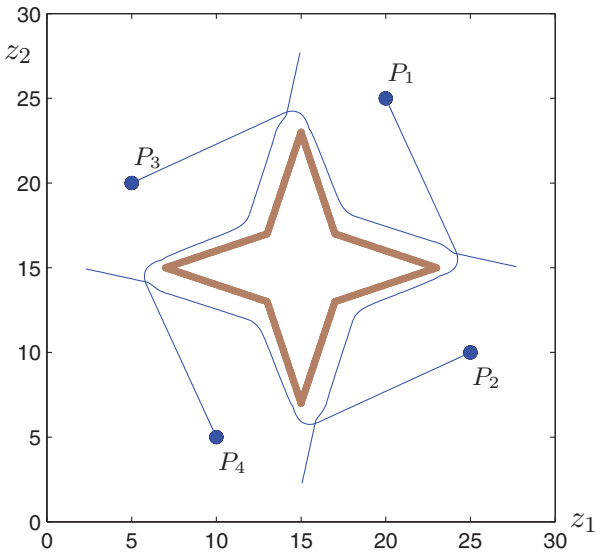


Fig. 7. Collision-free trajectories of four point-mass robots in the presence of line obstacles.

The controllers u_{i1} and u_{i2} are bounded and continuous over the domain

$$D_{i4} = \{(x_i, y_i) \in \mathbb{R}^2 : \mathbf{x}_i(0) \neq \mathbf{e}_i \cap (x_i - X_k)^2 + (y_i - Y_k)^2 > r_{p_i}^2 \text{ for } k = 1, 2, \ldots, m\}.$$

Example 4. *Figure 7 shows an interesting computer simulation with four point-masses in a workspace that contains line obstacles. The initial and target position of the robots and the parameters related to the line segments were a priori chosen to achieve a pseudo traffic situation. From the simulation, we see that the point masses avoid the line segments along its path to their targets.*

4.5. Multi-tasking

We now combine the avoidance schemes described in Sections 4.1 to 4.4 to form a generalized MPC scheme for the avoidance of multiple types of obstacles and convergence to the target. Therefore, we define ϵ_i as:

$$\epsilon_i = \tan^{-1}\left(\sum_{s=1}^{4} \frac{\alpha_{is}\beta_{is}}{R_{is}}\right). \tag{10}$$

With the form of ϵ_i, we see that the controllers u_{i1} and u_{i2} are given as

$$\begin{aligned} u_{i1} &= \frac{|v_0|}{\|\mathbf{x}_i(0) - \mathbf{e}_i\|}\left[\frac{(p_{i1} - x_i) - (p_{i2} - y_i)\sum_{s=1}^{4}\frac{\alpha_{is}\beta_{is}}{R_{is}}}{\sqrt{1 + \left(\sum_{s=1}^{4}\frac{\alpha_{is}\beta_{is}}{R_{is}}\right)^2}}\right] \\ u_{i2} &= \frac{|v_0|}{\|\mathbf{x}_i(0) - \mathbf{e}_i\|}\left[\frac{(p_{i2} - y_i) + (p_{i1} - x_i)\sum_{s=1}^{4}\frac{\alpha_{is}\beta_{is}}{R_{is}}}{\sqrt{1 + \left(\sum_{s=1}^{4}\frac{\alpha_{is}\beta_{is}}{R_{is}}\right)^2}}\right] \end{aligned} \tag{11}$$

https://doi.org/10.1017/S0263574714002070

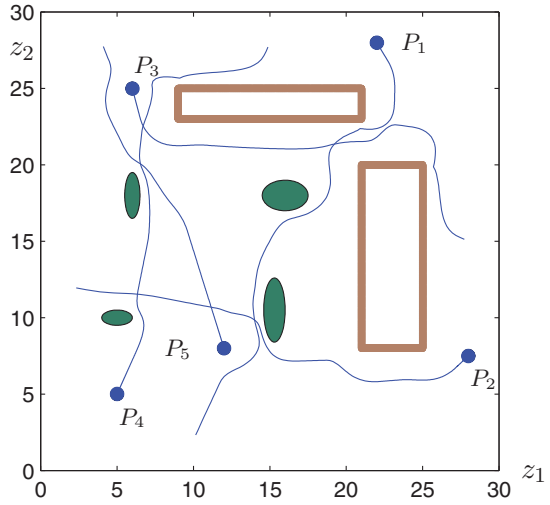


Fig. 8. Collision-free trajectories of P_i for $i = 1, 2, \ldots, 5$ in a obstacle-ridden WS.

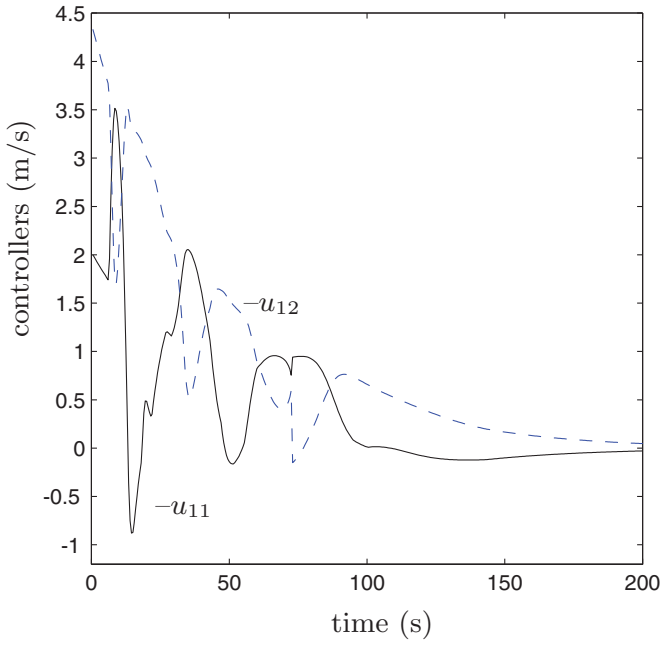


Fig. 9. Evolution of the control signals, u_{11} and u_{12} along the trajectory of P_1.

which are bounded and continuous at every point over the domain

$$D_i = \{(x_i, y_i) \in \mathbb{R}^2 : \mathbf{x}_i(0) \neq \mathbf{e}_i \cap R_{is} > 0 \text{ for } s = 1, 2, 3, 4\}.$$

Example 5. *The simulation shown in Fig. 8 is of five point-mass robots maneuvering in the workspace WS cluttered with fixed and moving obstacles with randomized sizes. We have considered a simple setup where each robot maneuvers from an initial to a final configuration, whilst avoiding obstacles of different types along the path. The rectangular obstacles made up of line segments may represent a building in real-life situation. Figure 9 shows the time evolution of the nonlinear controllers for P_1. We can see that eventually, at the center of the target the controllers became zero.*

4.6. Effect of noise

To evaluate the robustness of our current approach, we now look at the effect of noise that could be encountered in the sensor measurements. Since we are considering sensing zones around each obstacle, it suffices to include the noise components into the parameters α_{is} and β_{is} as follows:

$$\alpha_{is} = \begin{cases} 0, & \text{if } \hat{R}_{is} \geq d_{\max} \\ d_{\max} - \hat{R}_{is}, & \text{if } \hat{R}_{is} < d_{\max} \end{cases} \quad \text{and} \quad \beta_{is} = \begin{cases} 1, \text{ if } \hat{f}_{is} \leq 0 \\ -1, \text{ if } \hat{f}_{is} > 0 \end{cases}$$

where $\hat{R}_{is} = R_{is} + \sigma\mu_{is}(t)$ and $\hat{f}_{is} = f_{is} + \sigma\nu_{is}(t)$, for $s = 1, 2, 3, 4$. The terms $\sigma\mu_{is}(t)$ and $\sigma\nu_{is}(t)$ are the small disturbances in the sensor readings, the constant $\sigma \in [0, 1]$ is the noise level while $\mu_{is}(t)$ and $\nu_{is}(t)$ are time-dependent variables randomized between and including -1 and 1.

Example 6. *To explore the effect of noise in the simulation, we have regenerated the simulation of Fig. 8. The new simulations at different noise levels are shown in Fig. 10. The parameter σ is given in the captions of each subfigure while $\mu_{is}(t)$ and $\nu_{is}(t)$ were taken as random numbers in the interval $[-1, 1]$ for $s = 1, 2, 3, 4$ and $i = 1, 2, \ldots, 5$. We have also plotted the time evolution of the nonlinear controllers for P_1 for each of the simulations. We again see that eventually, at the center of the target the controllers became zero. Some minor changes are noticed in the trajectories of the point-masses and in the controllers.*

5. Stability Analysis

The new controllers u_{i1} and u_{i2} are bounded and continuous at every point over the domain

$$D_i = \{(x_i, y_i) \in \mathbb{R}^2 : \mathbf{x}_i(0) \neq \mathbf{e}_i \cap R_{is} > 0 \text{ for } s = 1, 2, 3, 4\}.$$

This gives rise to the following theorem:

Theorem 1. *The point $\mathbf{e}_i$ $(i = 1, 2, \ldots, n)$ is a global asymptotic stable equilibrium point of system (1).*

Proof. Consider the Lyapunov function

$$L = \sum_{i=1}^{n} L_i(\mathbf{x}_i)$$

where $L_i(\mathbf{x}_i) = \frac{1}{2}\|\mathbf{x}_i(t) - \mathbf{e}_i\|^2$, which is defined, continuous, positive and radially unbounded over the domain

$$D_i = \{\mathbf{x}_i(t) \in \mathbb{R}^2 : \mathbf{x}_i(0) \neq \mathbf{e}_i \cap R_{is} > 0 \text{ for } s = 1, 2, 3, 4\}.$$

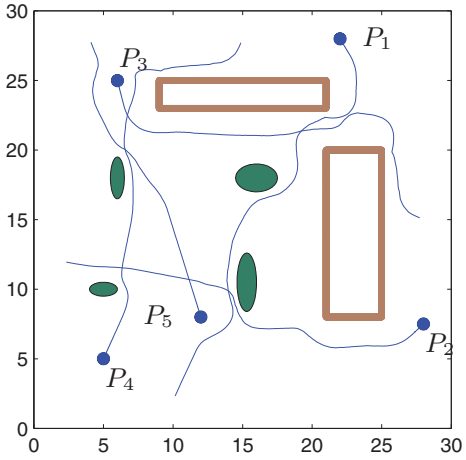
It is clear that L has first partial derivatives in the neighborhood $D_{(1)} = D_1 \cap D_2 \cap \cdots \cap D_n$ of the equilibrium point $\mathbf{e}_i$ of system (1). Moreover, in the region D_i, we see that $L_i(\mathbf{e}_i) = 0$ and $L_i(\mathbf{x}_i) > 0$ for all $\mathbf{x}_i \neq \mathbf{e}_i$.

Now, the time-derivative of $L_i(\mathbf{x}_i)$ along a trajectory of system (1) is given by

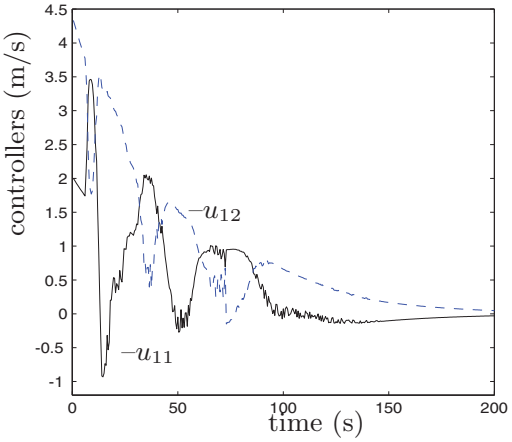
$$\dot{L}_i(\mathbf{x}_i) = -\sqrt{u_{i1}^2 + u_{i2}^2}\ \ \|\mathbf{x}_i(t) - \mathbf{e}_i\|\ \cos\epsilon_i.$$

Again, it is clear that in the region D_i, $\dot{L}_i(\mathbf{e}_i) = 0$ and $\dot{L}_i(\mathbf{x}_i) < 0$ for all $\mathbf{x}_i \neq \mathbf{e}_i$. Hence it can be concluded that the point $\mathbf{e}_i$ for $i \in \{1, 2, \ldots, n\}$ is a *global asymptotic stable* equilibrium point of system (1).

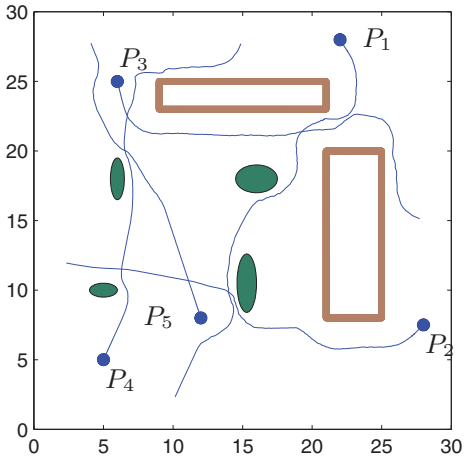
https://doi.org/10.1017/S0263574714002070



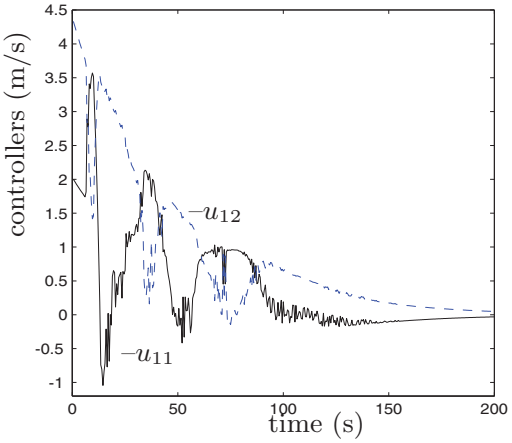
(a) Trajectories for $\sigma = 0.1$.



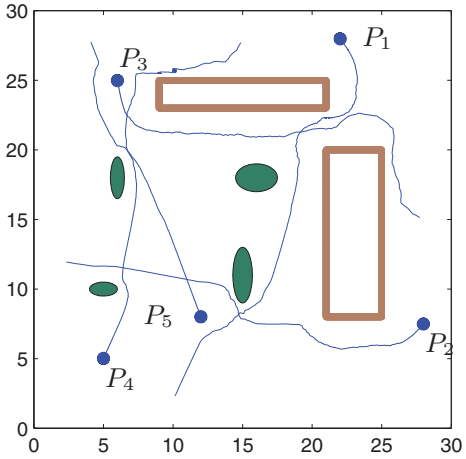
(b) Control signals, u_{11} and u_{12} for $\sigma = 0.1$.



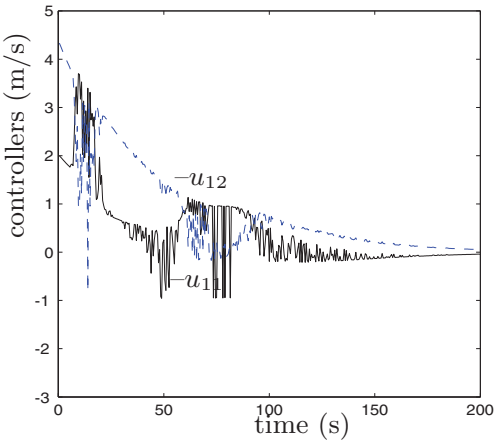
(c) Trajectories for $\sigma = 0.2$.



(d) Control signals, u_{11} and u_{12} for $\sigma = 0.2$.



(e) Trajectories for $\sigma = 0.5$.



(f) Control signals, u_{11} and u_{12} for $\sigma = 0.5$.

Fig. 10. Trajectories of P_i and the evolution of control signals for P_1 under the influence of noise.

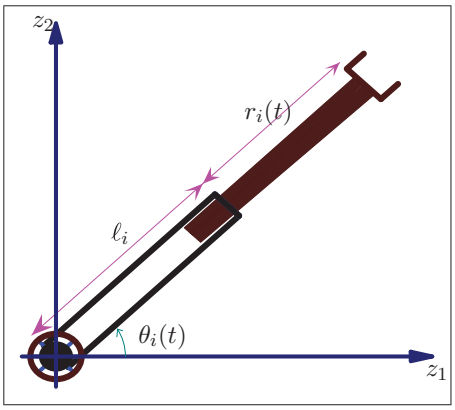


Fig. 11. Schematic representation of the ith planar (RP) manipulator in the z_1z_2 plane. (Adopted from[15])

6. Application: Two Planar Robot Arms in WS

We apply the approach to a system of two planar robot arms operating together in a common workspace WS. The robot arms have a translational joint and a rotational joint in the z_1z_2 plane as shown in Fig. 11. The arm consists of two links made up of uniform slender rods; the revolute first link with fixed length and the prismatic second link which caries the payload at the gripper.

With the help of Fig. 11, we assume:

(i) the planar robot arm is anchored at the point (a_i, b_i);
(ii) the first link has a fixed length ℓ_i;
(iii) the second link has length $r_i(t)$ at time t; and
(iv) the manipulator has angular position $\theta_i(t)$ at time t;
(v) the coordinate of the gripper is $(x_i(t), y_i(t))$.

Remark: We can express the position of the end-effector of the articulated manipulator arm completely in terms of the state variables $r_i(t)$ and $\theta_i(t)$ as:

$$x_i(t) = a_i + (\ell_i + r_i(t)) \cos\theta_i(t),$$
$$y_i(t) = b_i + (\ell_i + r_i(t)) \sin\theta_i(t).$$

The ith planar robot arm is governed by the following system of ODEs:

$$\left.\begin{aligned} \dot{r}_i(t) &= u_{i1}\cos\theta_i + u_{i2}\cos\theta_i, \\ \dot{\theta}_i(t) &= \frac{u_{i2}\cos\theta_i - u_{i1}\sin\theta_i}{\ell_i + r_i}, \\ r_i(0) &= \sqrt{(x_i(0) - a_i)^2 + (y_i(0) - b_i)^2} - \ell_i, \\ \theta_i(0) &= \text{atan2}\,(y_i(0) - b_i, x_i(0) - a_i)\,, \end{aligned}\right\} \tag{12}$$

for $i = 1, 2$. System (12) is a description of the instantaneous velocities of the ith planar robot arm. Here u_{i1} and u_{i2} are again classified as the controllers. We shall use the vector notation $\mathbf{x}_i = (r_i(t), \theta_i(t))$ to refer to the position of the ith planar robot arm in the z_1z_2 plane.

In the following subsections, we consider different types of obstacles that the system may encounter.

6.1. Mechanical singularities

From a practical viewpoint, the motion of the manipulators are restricted in the sense that the end-effector of the 2-link manipulator can not go inside the first link. Thus a circular region of radius r_i with center (a_i, b_i) that encloses the first link is treated as an *artificial obstacle* for the end-effector.

https://doi.org/10.1017/S0263574714002070

6.2. Fixed obstacles

If the workspace contains fixed obstacles, then it is important for the entire Link 2 to avoid the obstacle. That is, if the end-effecter wants to overcome an obstacle from the side of the obstacle then the second link must be pulled inside the first link. For simplicity, assume that the lth obstacle is a circular disk with center (o_{l1}, o_{l2}) and radius ro_l.

In order for the entire Link 2 to avoid a fixed obstacle, it is important that every point on this link avoids the obstacle. For the avoidance, we again utilize the MDT by Sharma in ref. [1]. In our case, we want the line segment (Link 2) to avoid a fixed obstacle.

Let $(x_{il}^{(1)}, y_{il}^{(1)})$ be a point on the second link that is closest to the lth fixed obstacle. It can be shown that

$$x_{il}^{(1)} = \left(\ell_i + \lambda_{il}^{(1)}\right)\cos\theta_i, \qquad y_{il}^{(1)} = \left(\ell_i + \lambda_{il}^{(1)}\right)\sin\theta_i,$$

where $\lambda_{il}^{(1)} = o_{l1}\cos\theta_i + o_{l2}\sin\theta_i - \ell_i$. Note that $\lambda_{il}^{(1)} \in [0, r_i]$. Thus if $o_{l1}\cos\theta_i + o_{l2}\sin\theta_i - \ell_i < 0$, then we take $\lambda_{il}^{(1)} = 0$ and if $o_{l1}\cos\theta_i + o_{l2}\sin\theta_i - \ell_i > r_i$, then we take $\lambda_{il}^{(1)} = r_i$. We further define

$$R_{il}^{(1)} = \sqrt{\left(x_{il}^{(1)} - o_{l1}\right)^2 + \left(y_{il}^{(1)} - o_{l2}\right)^2} - ro_l$$

be the distance from the center of the lth obstacle to the point $(x_{il}^{(1)}, y_{il}^{(1)})$.

6.3. Moving obstacles

Since the two robots are working in the same workspace, each will be treated as a moving obstacle for the other. Thus each link of the jth manipulator becomes a moving obstacle for the ith manipulator. For the avoidance of the jth manipulator, it is necessary for the end-effector of ith manipulator to avoid the closest point on the jth manipulator. We again use MDT here. Suppose $(x_{ij}^{(2)}, y_{ij}^{(2)})$ is a point on a link of the jth manipulator that is closest to the end-effector of the ith manipulator, then it can be shown that

$$x_{ij}^{(2)} = a_j + \lambda_{ij}^{(2)}\cos\theta_j, \qquad y_{ij}^{(2)} = b_j + \lambda_{ij}^{(2)}\sin\theta_j,$$

where $\lambda_{ij}^{(2)} = (x_i - a_j)\cos\theta_j + (y_i - b_j)\sin\theta_j$. Note that $\lambda_{ij}^{(2)} \in [0, l_j + r_j]$. Thus if $(x_i - a_j\cos\theta_j + (y_i - b_j)\sin\theta_j < 0$, then we take $\lambda_{ij}^{(2)} = 0$ and if $(x_i - a_j\cos\theta_j + (y_i - b_j)\sin\theta_j > l_j + r_j$, then we take $\lambda_{ij}^{(2)} = l_j + r_j$. For $i, j = 1, 2$ and $i \neq j$, we further define

$$R_{ij}^{(2)} = \sqrt{\left(x_i - x_{ij}^{(2)}\right)^2 + \left(y_i - y_{ij}^{(2)}\right)^2}$$

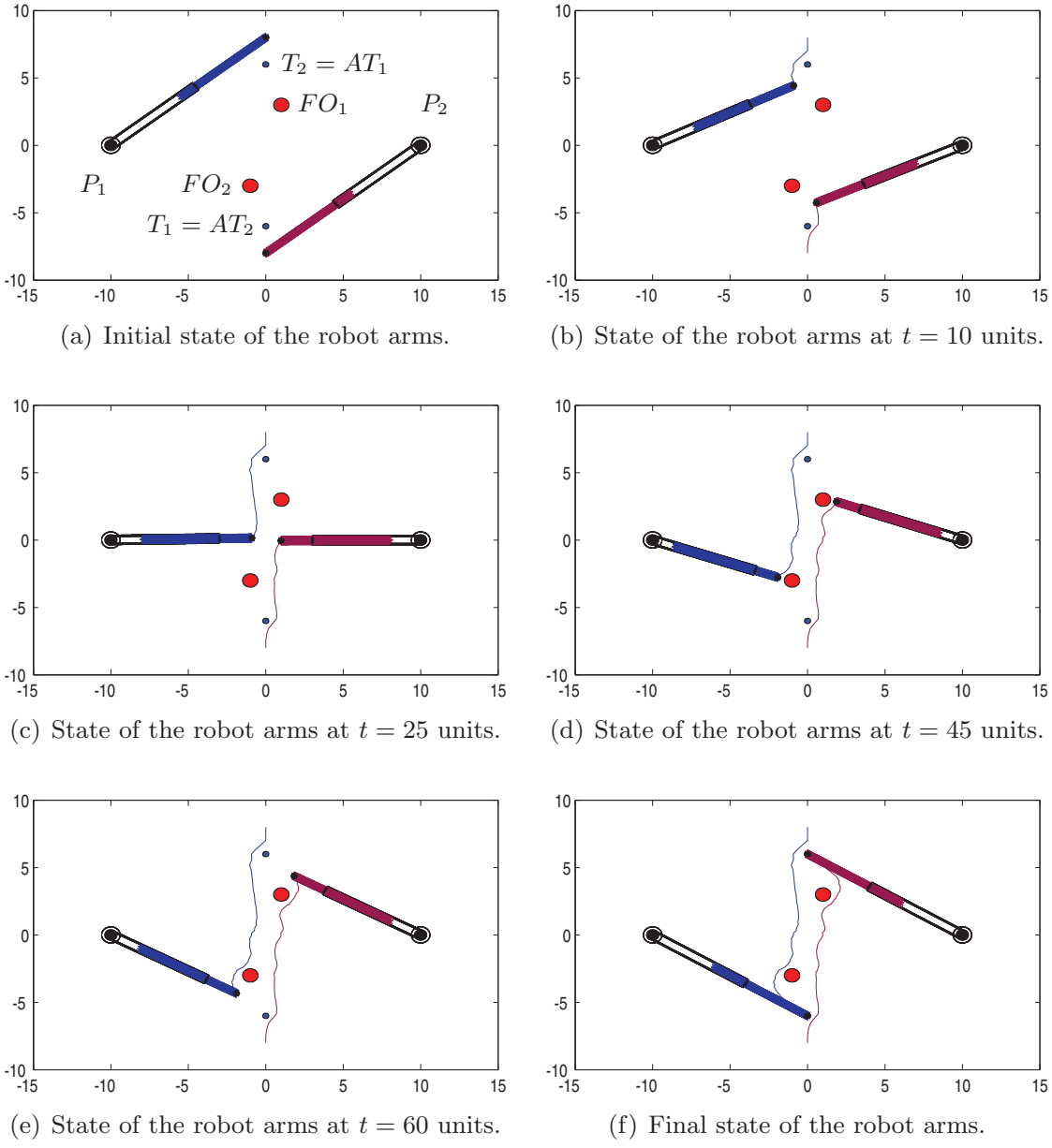
be the distance between the points $(x_{ij}^{(2)}, y_{ij}^{(2)})$ and (x_i, y_i).

6.4. Design of controllers

Taking into account the different types of obstacles discussed in Sections 6.1 to 6.3, we now design the control laws. Let $\mathbf{e}_i = (p_{i1}, p_{i2})$ be the target position of the ith end-effector. Consider $\hat{R}_i = \min(r_i, R_{il}^{(1)}, R_{ij}^{(2)}) + \sigma\mu_i(t)$ and

$$\alpha_i = \begin{cases} 0, & \text{if } \hat{R}_i \geq d_{\max} \\ d_{\max} - \hat{R}_i, & \text{if } \hat{R}_i < d_{\max} \end{cases} \quad \text{and} \quad \beta_i = \begin{cases} 1, \text{ if } \hat{f}_i \leq 0 \\ -1, \text{ if } \hat{f}_i > 0 \end{cases}$$

https://doi.org/10.1017/S0263574714002070



(a) Initial state of the robot arms.

(b) State of the robot arms at $t = 10$ units.

(c) State of the robot arms at $t = 25$ units.

(d) State of the robot arms at $t = 45$ units.

(e) State of the robot arms at $t = 60$ units.

(f) Final state of the robot arms.

Fig. 12. Snapshots of the trajectories of the robot arms traced by their end-effectors.

where $\hat{f}_i = p_{i1} \sin\theta_i - p_{i2} \cos\theta_i + \sigma\nu_i(t)$, $\sigma\mu_i(t)$ and $\sigma\nu_i(t)$ are the noise components. Then we design the controllers u_{i1} and u_{i2} as

$$\left.\begin{aligned} u_{i1} &= \frac{|v_0|}{\|\mathbf{x}_i(0) - \mathbf{e}_i\|} \left[\frac{(p_{i1} - (\ell_i + r_i)\cos\theta_i)R_i - (p_{i2} - (\ell_i + r_i)\sin\theta_i)\alpha_i\beta_i}{\sqrt{\alpha_i^2 + R_i^2}} \right] \\ u_{i2} &= \frac{|v_0|}{\|\mathbf{x}_i(0) - \mathbf{e}_i\|} \left[\frac{(p_{i2} - (\ell_i + r_i)\sin\theta_i)R_i + (p_{i1} - (\ell_i + r_i)\cos\theta_i)\alpha_i\beta_i}{\sqrt{\alpha_i^2 + R_i^2}} \right] \end{aligned}\right\} \tag{13}$$

We further note that the controllers are bounded and continuous at every point over the domain

$$D_i = \{(x_i, y_i) \in \mathbb{R}^2 : \mathbf{x}_i(0) \neq \mathbf{e}_i \cap R_i > 0\} \text{ for } i = 1, 2.$$

https://doi.org/10.1017/S0263574714002070

Example 7. *To illustrate the effectiveness of the proposed solution, this example involves a virtual situation wherein two planar robot arms have to move from their initial to final states whilst avoiding collisions and obstacles. The noise parameters taken here are* $\sigma = 0.3$ *while* $\mu_{is}(t)$ *and* $\nu_{is}(t)$ *are random numbers in the interval* $[-1, 1]$. *The iterative motion of the arms is shown in Fig. 12. We note here that when the end-effecter overcomes an obstacle from the side of the obstacle, the second link (Link 2) is pulled inside the first link (Link 1) ensuring that the entire arm avoids the obstacle. In the final maneuver the translational arm is pulled out so that the end-effector reaches the target.*

7. Concluding Remarks

The paper presents a simple yet systematic and robust scheme for solving the motion planning and control problem of multiple point-mass robots. A tailored velocity algorithm is used to drive the robots towards its goal at all times and render it stationary once it reaches this goal. Then with a careful definition of the turning angle ϵ_i, we generate the control laws so that the robots can avoid various types of obstacles along their paths.

The control laws proposed in this paper also ensure an asymptotic stability of the system. This is proven using the Direct Method of Lyapunov. While we have proved the asymptotic stability of the system, computer simulations using point-mass robots and anchored 2-link (RP) manipulators highlight numerically the stabilization property of the system. In addition, we have studied the effect of noise in simulations showing the robustness of the system.

Future work will consider real-time experiments by incorporating the proposed controllers into real robots. Other work will involve motion planing and control of nonholonomic mechanical systems in literature. The car-like robots, tractor-trailor systems and the mobile manipulators are such examples.

Acknowledgements

The authors would like to sincerely thank the referees and Professor Jun-Hong Ha of Korea University of Technology and Education, Korea, for their helpful comments which have led to an improvement in the content and presentation of the paper.

References

1. B. Sharma, New Directions in the Applications of the Lyapunov-based Control Scheme to the Findpath Problem *Ph.D. Thesis* (University of the South Pacific, Suva, Fiji Islands, July 2008). PhD Dissertation, http://www.staff.usp.ac.fj/~sharma_b/Bibhya_PhD.pdf.
2. B. Sharma, J. Vanualailai and S. Singh, "Tunnel passing maneuvers of prescribed formations," *Int. J. Robust Nonlinear Control* (2012).
3. W. Burgard, M. Moors, C. Stachniss and F. E. Schneider, "Coordinated multi-robot exploration," *IEEE Trans. Robot.* **21**(3), 376–386 (2005).
4. H. Hu, P. W. Tsui, L. Cragg and N. Völker, "Architecture for multi-robot cooperation over the internet," *Int. J. Integr Comput.-Aided Eng.* **11**(3), 213–226 (2004).
5. H. Yamaguchi, "A distributed motion coordination strategy for multiple nonholonomic mobile robots in cooperative hunting operations," *Robot. Auton. Syst.* **43**(4), 257–282 (2003).
6. F. Arrichiello, Coordination Control of Multiple Mobile Robots *Ph.D. Thesis* (Universita Degli Studi Di Cassino, Cassino, Italy, November 2006). PhD Dissertation.
7. D. M. Dawson, E. Zergeroglu, W. E. Dixon and F. Zhang, "Robust tracking and regulation control for mobile robots," *Int. J. Robust Nonlinear Control* **10**(4), 199–216 (2000).
8. E. Rimon, "Exact robot navigation using artificial potential functions," *IEEE Trans. Robot. Autom.* **8**(5), 501–517 (1992).
9. H. G. Tanner, S. Loizou and K. J. Kyriakopoulos, "Nonholonomic navigation and control of cooperating mobile manipulators," *IEEE Trans. Robot. Autom.* **19**(3), 53–64 (2003).
10. R. C. Arkin, "Motor schema-based mobile robot navigation," *Int. J. Robot. Res.* **8**(4), 92–112 (1989).
11. M. D. Adams, H. Hu and P. J. Probert, "Towards a Real-Time Architecture for Obstacle Avoidance and Path Planning in Mobile Robots," *Proceedings. IEEE International Conference on Robotics and Automation*, Vol. 4 (1990).
12. P. Khosla and R. Volpe, "Superquadric Artificial Potential for Obstacle Avoidance and Approach," *Proceedings of the IEEE International Conference on Robotics and Automation* (1988) pp. 1778–1784.

13. J. Vanualailai, J-H. Ha and B. Sharma, "An asymptotically stable collision-avoidance system," *Int. J. Non-Linear Mech.* **43**(9), 925–932 (2008).
14. B. Sharma, A. Prasad and J. Vanualailai, "A collision-free algorithm of a point-mass robot using neural networks," *J. Artif. Intell.* **3**(1), 49–55 (2012).
15. A. Prasad, B. Sharma and J. Vanualailai, "Motion planning and control of autonomous robots in a two-dimensional plane," *World Acad. Sci. Eng. Technol.* **6**(12), 1163–1168 (2012).

https://doi.org/10.1017/S0263574714002070