A behaviour-based control approach for the trajectory tracking of an underactuated planar capsule robot

M. Nazmul Huda*, Hongnian Yu** and Shuang Cang***

Faculty of Science and Technology, Bournemouth University, UK.
**nhuda@bournemouth.ac.uk, **Corresponding author, yuh@bournemouth.ac.uk*

****School of Tourism, Bournemouth University, UK.*
scang@bournemouth.ac.uk

Abstract

A miniature capsule robot (capsubot) - which has no external moving parts whereas a conventional robot has legs and/or wheels - is suitable for in-vivo applications, engineering diagnosis and pipe inspection. This study addresses the trajectory-tracking problem of an underactuated planar capsubot. A combining piece-wise and behaviour-based control algorithm is proposed for trajectory tracking. The paper also proposes four motion behaviours, four switching behaviours and one stationary behaviour. A selection algorithm for behavior-based control and rules for inner mass (IM) motion control in all the behaviours are developed. The partial feedback linearization control is used for low-level IM motion control while the piece-wise and behaviour based control is used for the capsubot trajectory tracking control.

1. Introduction

Control of underactuated mechanical systems [1] is an active research area because of their simple structures, medical and industrial applications. A capsubot belongs to a class of underactuated system which works on the principle of internal reaction force and, hermetically sealable and limbless - the examples are a pendulum on a cart [2], a 1D capsubot [3] etc. Thus a miniature capsubot is suitable for in-vivo applications such as medical diagnosis (e.g. capsule endoscopy) & treatments (e.g. drug delivery) whereas a micro/miniature legged robot [4] may injure the internal soft tissues by the sharp edges of the legs. It is also suitable for engineering diagnosis specially underground pipe inspection as the outer structure of the capsubot can be designed to match the desired pipe structure. Recently the research of micro/miniature robots, e.g. legged [4] and legless robots [5, 6, 7, 8] including capsule robots [3, 9, 10] have attracted the attention of researchers because of their potential applications.

The capsubot propulsion principle was analyzed in [11] from the viewpoint of physics. The motion generation of a single mass capsubot was explained in [9] based on a four-step velocity profile which was, fast motion for the first two steps and slow motion in the last two steps. In [10], the motion

generation of the capsubot was explained on the basis of a seven-step velocity profile which is, fast motion in the first three steps and slow motion in the rest of the steps. Motion of a single mass capsubot was explained in [3] based on a four-step acceleration profile and a stand-alone prototype was developed. However trajectory tracking which is the primary requirement of this type of underactuated systems was not considered in the literature. This paper addresses 2D trajectory tracking control of a capsubot. It proposes a combined piece-wise and behaviour-based [12, 13, 14] algorithm for planar trajectory tracking of a 2D capubot.

Main contributions of the paper are 1) defining various basis behaviours for the 2D capsubot (section 2.3); 2) proposing a trajectory tracking control algorithm by combining piece-wise and behaviour-based control for the trajectory tracking control (section 3); 3) developing a selection algorithm for proper selection of the behavior-set (section 4.2.2); 4) describing the data set creation, selection of profile parameters, tuning the piece time and, modifying the desired velocities (sections 4.2.3 - 4.2.6); 5) proposing the rules for implementing each behaviour (section 4.2.7).

2. System Description and Defining the Behaviours

2.1. System Description: 2D Capsubot Model:

The 2D capsubot considered in this paper has a parallelepiped shape shown in Fig. 1(a). The 2D capsubot is an underactuated system as it has four degrees of freedom but only two control inputs. Two IMs (inner masses) are placed in the hollow spaces within the capsubot. The hollow spaces are identical and placed symmetrically within the capsubot. IMs can move along the hollow spaces. By controlling the movements of IMs, the capsubot can move on a plane. F_{m_i} force is applied on the IM_i along the hollow space and creates a motion whereas f_{m_i} is the friction force. IM_i applies equal and opposite forces on the capsubot. The sources of the forces which are not shown in the figure could be linear motors as used in [3, 15]. From Fig. 1(a) the capsubot dynamic model is:

$$F_{m_i} - f_{m_i} = m_i \ddot{x}_{mi} \quad \forall \ i = 1, 2 \tag{1}$$

$$\sum F_x = M\ddot{x} = (F_r - f_M)cos(\phi) = (-F_{m_1} + f_{m_1} - F_{m_2} + f_{m_2} - f_M)cos(\phi) \tag{2}$$

$$\sum F_y = M\ddot{y} = (F_r - f_M)sin(\phi) = (-F_{m_1} + f_{m_1} - F_{m_2} + f_{m_2} - f_M)sin(\phi) \tag{3}$$

$$\sum M_G = I\ddot{\phi} = M_r - M_f = (-F_{m_2} + f_{m_2})l_2 - (-F_{m_1} + f_{m_1})l_1 - M_f \tag{4}$$

where $\ddot{x}_{mi}$ is the acceleration of IM_i; x, y and ϕ are generalised coordinates of the capsubot with respect to $O(X_O, Y_O)$; m_i and M are the IM_i mass and capsubot mass respectively; F_r is the total reaction forces of the IMs on the capsubot, M_r is the total moment due to reaction forces of the IMs on the capsubot about z-axis through the capsubot mass centre; l_i is the perpendicular distance

of the direction of force F_{m_i} from the axis of rotation; f_M is the friction on the capsubot by the surface of motion - $f_M = 0$ if $F_r = 0$ and $|f_M|$ increases when $|F_r|$ increases with a maximum value of $f_{MM} = sgn(\dot{r})\mu Mg$; $\dot{r}$ is the capsubot linear velocity, μ=translational friction coefficient, g=gravitational constant; M_f is the capsubot frictional moment about z-axis through the capsubot mass centre - $M_f = 0$ if $M_r = 0$ and $|M_f|$ increases when $|M_r|$ increases with a maximum value of $M_{fM} = sgn(\dot{\phi})\frac{2}{3}\mu_r Mg(r_2 + \frac{wl-\pi r_2^2}{\pi r_1})$, $r_2 = \frac{w}{2}$, $r_1 = \frac{\sqrt{l^2+w^2}}{2}$ [15]; μ_r = rotational friction coefficient, l and w are the length and width of the capsubot respectively, $I = \frac{1}{12}M(l^2 + w^2)$ is the moment of inertia of the capsubot about z-axis through the capsubot mass centre.

2.2. System Description: Motion Generation and Switching

The capsubot has one switching mode and two motion modes: linear motion mode and rotational motion mode depending on the forces applied on the IMs. For easy implementation, we design the capsubot with $m_1 = m_2 = m$ and $l_1 = l_2$. The mass centre of the capsubot is assumed to stay at a fixed point within the capsubot and the natural mass centre of a parallelepiped i.e. G of Fig. 1.

In the linear and rotational motion modes the IMs follow a four-step acceleration profile given below. One example of the acceleration profile is shown in Fig. 2(a).

$$\ddot{x}_{mid} = \begin{cases} a_{mi1} & 0 \le t < t_{i1};\ \forall\, i = 1,2 \quad Step\ 1:\ Onward\ journey\ of\ IM_i \\ a_{mi2} & t_{i1} \le t < t_{i2};\ \forall\, i = 1,2 \quad Step\ 2:\ Onward\ journey\ of\ IM_i \\ a_{mi3} & t_{i2} \le t < t_{i3};\ \forall\, i = 1,2 \quad Step\ 3:\ Return\ journey\ of\ IM_i \\ a_{mi4} & t_{i2} \le t < t_{i4};\ \forall\, i = 1,2 \quad Step\ 4:\ Return\ journey\ of\ IM_i \end{cases} \tag{5}$$

After choosing a_{mi1} to a_{mi4} based on the desired capsubot velocities, t_{i1} to t_{i4} can be found as:

$$t_{i1} = \frac{|v_{mi1}|}{|a_{mi1}|}; \quad t_{i2} = t_{i1} + \frac{|v_{mi1}|}{|a_{mi2}|}; \quad t_{i3} = t_{i2} + \frac{|v_{mi3}|}{|a_{mi3}|}; \quad t_{i4} = t_{i3} + \frac{|v_{mi3}|}{|a_{mi4}|} \tag{6}$$

where v_{mi1} and v_{mi3} are the IM_i velocities at the end of steps 1 and 3 respectively.

In the switching mode the IMs follow a two-step acceleration profile given below:

$$\ddot{x}_{mid} = \begin{cases} a_{msi1} & 0 \le t < t_{i1};\ \forall\, i = 1\, and/or\, 2 \quad Step\ 1:\ Onward\ journey\ of\ IM_i \\ a_{msi2} & t_{i1} \le t < t_{i2};\ \forall\, i = 1\, and/or\, 2 \quad Step\ 2:\ Onward\ journey\ of\ IM_i \end{cases} \tag{7}$$

For simplicity, we choose $a_{msi1} = -a_{msi2}$. Then t_{i1} and t_{i2} can be found as:

$$t_{i1} = \sqrt{\frac{2k}{|a_{msi1}|}}; \quad t_{i2} = 2t_{i1} \tag{8}$$

where k = maximum stroke length of IM_i

Linear Motion Mode: If forces of same magnitude and direction are applied to both the IMs i.e. $F_{m_1} = F_{m_2}$ then $M_r = 0$ and $\sum M_G = 0$ in (4), and $F_r \neq 0$ in (2) and (3). Thus if the capsubot

has a zero initial velocity and $|F_r| > |f_{MM}|$, the capsubot starts a linear motion. Here $\ddot{x}_{m1d} = \ddot{x}_{m2d}$. Capsubot performs linear motion in steps 2, 3 and part of step 4.

Rotational Motion Mode: If forces of same magnitude but opposite directions are applied to the IMs i.e. $F_{m_1} = -F_{m_2}$ then $M_r \neq 0$ in (4), and $F_r = 0, \sum F_x = \sum F_y = 0$ in (2) and (3). Thus if the capsubot has a zero initial velocity and $|M_r| > |M_{fM}|$, the caspubot starts a rotational motion. Here $\ddot{x}_{m1d} = -\ddot{x}_{m2d}$. Capsubot performs rotational motion in steps 2, 3 and part of step 4.

Switching Mode: Capsubot uses this mode to switch from one motion to another. In this mode the IM/IMs changes/change its/their position from one end to other but the capsubot remains stationary. Here the forces applied on the IMs are small enough so that $|F_r| < |f_{MM}|$ and $|M_r| < |M_{fM}|$. Thus the IMs' accelerations maintain following constraint: $|\ddot{x}_{mid}| < min(\frac{|f_{MM}|}{2m_i}, \frac{|M_{fM}|}{2m_i l_i})$

2.3. System Description: Basis Behaviours

We define the following nine basis behaviours based on the above motion and switching modes. These behaviours are followed for trajectory tracking.

1. Forward (FW) linear motion: The IMs' initial and final positions are the rear end of the capsubot; the capsubot moves forward.
2. Backward (BW) linear motion: The IMs' initial and final positions are the front end of the capsubot; the capsubot moves backward.
3. Clock-wise (CW) rotational motion: The initial and final positions of IM_1 are the rear end of the capsubot and of IM_2 are the front end; the capsubot rotates clockwise.
4. Counter clock-wise (CCW) rotational motion: The initial and final positions of IM_1 are the front end of the capsubot and of IM_2 are the rear end; the capsubot rotates clockwise.
5. Switching to FW linear motion: Using this behaviour both the IMs reach to the rear end of the capsubot and ready to start FW linear motion.
6. Switching to BW linear motion: Using this behaviour both the IMs reach to the front end of the capsubot and ready to start BW linear motion..
7. Switching to CW rotational motion: Using this behavior IM_1 and IM_2 reach to the rear and front end of the capsubot respectively and ready to start CW rotational motion.
8. Switching to CCW rotational motion: Using this behavior IM_1 and IM_2 reach to the front and rear end of the capsubot respectively and ready to start CCW rotational motion.
9. Stationary: Both the IMs remain stationary and thus the capsubot also remains stationary.

2.4. Reference Frame Allocation

We assign a fixed reference frame $O(X_O, Y_O)$ and following local frames while the robot moves as shown in Fig. 1. We assign two local frames: $R_j(X_{R_j}, Y_{R_j})$ and $L_j(X_{L_j}, Y_{L_j})$ on the mass centre of

the capsubot. The robot performs only one behaviour at a time. When the capsubot needs to move from one point to another, it uses rotational motion to align itself with the straight line joining current position and destination; then it uses linear motion to move to the destination. When the capsubot rotates R_j remains stationary with respect to O and, L_j moves with the capsubot. Then L_j become stationary with respect to O and the capsubot performs linear motion. When the robot moves to the next destination two more local frames are assigned. The current orientation and position of the capsubot with respect to O are:

$$\phi_j = \phi_{j-1} + \phi_{Mj}; \quad x_j = x_{j-1} + x_{Mj}cos(\phi_j); \quad y_j = y_{j-1} + x_{Mj}sin(\phi_j) \tag{9}$$

where ϕ_{Mj} is the rotation of the capsubot with respect to R_j and x_{Mj} is the translation of the capsubot with respect to L_j.

When the capsubot switches from one motion behaviour to another it uses the switching mode while the capsubot remains stationary but the IM/IMs moves/move.

Linear Motion Mode: In the local frame (L_j) the motion equations (1)-(4) become:

$$F_{m_1} - f_{m_1} = m_1\ddot{x}_{m1} = F_{m_2} - f_{m_2} = m_2\ddot{x}_{m2}; \quad \sum F_x = M\ddot{x}_M = -2F_{m_1} + 2f_{m_1} - f_M \tag{10}$$

where $f_{m_i} = sgn(\dot{x}_{mi} - \dot{x}_M)\mu_i m_i g \ \forall \ i = 1, 2$ and $f_M = sgn(\dot{x}_M)\mu Mg$, x_{mi} and x_M are the displacement of the IM_i and the capsubot respectively measured in the local frame (L_j).

The average linear velocity of the capsubot $\bar{\dot{x}}_M$ is:

$$\bar{\dot{x}}_M = \frac{x_M}{t_l} \tag{11}$$

where x_M is the linear displacement of the capsubot in cycle time t_l.

$$x_M = \frac{v_{M2}^2}{2a_{M2}} + \frac{v_{M3}^2 - v_{M2}^2}{2a_{M3}} - \frac{v_{M3}^2}{2a_{M4}}; \quad t_l = t_{i4} = \frac{|v_{mi1}|}{|a_{mi1}|} + \frac{|v_{mi1}|}{|a_{mi2}|} + \frac{|v_{mi3}|}{|a_{mi3}|} + \frac{|v_{mi3}|}{|a_{mi4}|} \tag{12}$$

where a_{Mq} is the capsubot acceleration in step q (where $q = 1, 2, 3and4$), v_{M2} and v_{M3} are the capsubot velocities at the end of steps 2 and 3 respectively.

$$v_{M2} = a_{M2}(t_{i2} - t_{i1}); \ \ v_{M3} = a_{M2}(t_{i2} - t_{i1}) + a_{M3}(t_{i3} - t_{i2})$$
$$a_{Mk} = \frac{-m_1 a_{m1k} - m_2 a_{m2k} - \mu Mg}{M}; \ \ k = 1, 2, 3, 4$$

Rotational Motion Mode: In the local frames (L_j, R_j) the motion equations (1)-(4) become:

$$F_{m_i} - f_{m_i} = m_i\ddot{x}_{mi} \quad \forall \ i = 1, 2; \quad \sum M_G = I\ddot{\phi}_M = (2F_{m_1} - 2f_{m_1})l_1 - M_f \tag{13}$$

where $f_{m_i} = sgn(\dot{x}_{mi})\mu_i m_i g$, $M_{fM} = sgn(\dot{\phi}_M)\frac{2}{3}\mu_r Mg(r_2 + \frac{wl - \pi r_2^2}{\pi r_1})$ [15], ϕ_M is the orientation of the capsubot in the local frame.

The average angular velocity of the capsubot $\bar{\dot{\phi}}_{Mi}$ is:

$$\bar{\dot{\phi}}_{Mi} = \frac{\phi_{Mi}}{t_r} \tag{14}$$

where ϕ_{Mi} is the angular displacement of the capsubot in cycle time t_r.

$$\phi_M = \frac{\omega_{M2}^2}{2\alpha_{M2}} + \frac{\omega_{M3}^2 - \omega_{M2}^2}{2\alpha_{M3}} - \frac{\omega_{M3}^2}{2\alpha_{M4}}; \quad t_r = t_{i4} = \frac{|v_{mi1}|}{|a_{mi1}|} + \frac{|v_{mi1}|}{|a_{mi2}|} + \frac{|v_{mi3}|}{|a_{mi3}|} + \frac{|v_{mi3}|}{|a_{mi4}|} \tag{15}$$

where α_{Mq} is the capsubot angular acceleration in step q (where $q = 1, 2, 3 and 4$), ω_{M2} and ω_{M3} are the capsubot angular velocities after steps 2 and 3 respectively.

$$\omega_{M2} = \alpha_{M2}(t_{i2} - t_{i1}); \quad \omega_{M3} = \alpha_{M2}(t_{i2} - t_{i1}) + \alpha_{M3}(t_{i3} - t_{i2})$$

$$\alpha_{Mk} = (m_2 a_{m2k} l_2 - m_1 a_{m1k} l_1 - M_{fk})/I \quad k = 1, 2, 3, 4$$

Switching Mode: In the local frame (L_j) the motion equations (1) to (4) become:

$$F_{m_i} - f_{m_i} = m_i \ddot{x}_{mi} \quad \forall \ i = 1, 2 \quad where \ \ f_{m_i} = sgn(\dot{x}_{mi}) \mu_i m_i g$$

3. Proposed Trajectory Tracking Algorithm

Let the capsubot follow the planar position trajectory shown in Fig. 2(b) which is a sinusoidal trajectory. This trajectory reflects necessary complexity to test the performance of the proposed trajectory tracking algorithm. We propose the following algorithm (trajectory tracking control algorithm) to solve the trajectory tracking problem with the details in section 4.

Step 1: Generating trajectory piece: Divide the trajectory into small pieces as shown in Fig. 2(c), and compute the desired angular and linear velocities to track each piece.

Step 2: Behaviour-based control: A behaviour-based control approach tracks each piece from step 1.

Step 2.1: Behaviour-sets: Define nine basis behaviours. Several behaviour-sets (A to I in Fig. 3) comprising one or more basis behaviours are formed. These behaviour-sets include all necessary combinations of behaviours to track each piece of trajectory.

Step 2.2: Selection algorithm: A selection algorithm shown in Fig. 4 is used to select appropriate behaviour-set for each trajectory piece.

Step 2.3: Data-base creation: To track the trajectory, change the capsubot velocity by tuning the acceleration parameters of the IMs. Hence we create a data-base by computing capsubot linear and angular velocities for different profile parameters to feed into step 2.4.

Step 2.4: Selection of profile parameters: The desired velocity is compared with the data-base and the appropriate profile parameters are selected. The acceleration sets for which (22) and (23) give the minimum value, are selected for rotational and linear motion modes respectively. To switch among various motion modes, switching modes are used.

Step 2.5: Tuning the piece time: Tune the piece-time based on the selected parameters.

Step 2.6: Modification of the desired angular and linear velocities: The desired linear and angular velocities for each piece are modified based on the projected position of the capsubot before the start of the tracking of the piece.

Step 2.7: Rules for behaviours: Develop rules to implement behaviours of selected behaviour-sets.

Step 3: Low-level control: IMs movements for each behaviour is performed using partial feedback linearization control.

Summary of the Algorithm:. Step 1 is used to generate pieces from the desired trajectory. Step 2.1 is used to define behaviours and behaviour-sets. Step 2.2 is used to select appropriate behavour-set to track the trajectory in a piece. Then in step 2.4 appropriate profile parameters are selected for the selected behaviour-set. These profile parameters are the desired accelerations of the IMs $\ddot{x}_{mid}, \forall\, i = 1, 2$. In step 3 the low-level IMs controller tracks the desired IMs accelerations $\ddot{x}_{mid}$ and eventually track the capsubot trajectory in a piece. The process will be iteratively performed for the rest of the pieces.

4. Methods for Implementing the Proposed Trajectory Tracking Algorithm

4.1. Step 1: Generating Trajectory Piece

The trajectory tracking is performed in a piece-wise manner. The desired trajectory of Fig. 2(b) is divided into small pieces with a piece-time T, as shown in Fig. 2(c). The capsubot follows the straight lines connecting the start and end points of the pieces. Firstly the capsubot aligns itself with the straight line by using one of the rotational behviours i.e. the capsubot corrects its steering angle. Then the capsubot uses one of the switching behaviours to switch from rotational to linear motion mode. Finally the capsubot travels the distance of the straight line using one of the linear behaviours.

The smaller the piece-time, the smoother the followed trajectory. However as the capsubot may need to complete behaviour-set comprising upto four behaviours - switching to rotation, rotation, switching to linear and linear motion - to track the trajectory in a piece, the capsubot should satisfy:

$$T \geq t_s + t_{rm} + t_s + t_{lm} \tag{16}$$

where t_s = time to complete the switching cycle, t_{rm} = maximum time to complete a rotation cycle, t_{lm} = maximum time to complete a linear cycle.

4.2. Step 2: Tracking using Behaviour-Based Control

4.2.1. Step 2.1: Behaviour-sets:

We have defined nine basis behaviours in section 2.3. A to I shown in Fig. 3 are all the possible behaviour-sets to follow certain piece of trajectory. E is used when the capsubot doesn't change its

position and orientation in the trajectory piece. A or B is used when the capsubot only changes its orientation whereas C or D is used when it only changes its position in the trajectory piece. F, G, H or I is used when the capsubot changes both of its position and orientation in the trajectory piece.

4.2.2. Step 2.2: Selection algorithm:

The selection algorithm shown in Fig. 4 is used to select the right behaviour-set - A to I - to track each trajectory piece. At the beginning of the tracking the IMs are placed at the rear end of the capsubot. The variable P is used to keep record of the behaviour of the capsubot - P = 1, 2, 3 or 4 means the previous behaviour executed is FW linear, BW linear, CCW rotational or CW rotational respectively. The rules developed in section 4.2.7 are used to implement behaviours of the selected behaviour-sets.

4.2.3. Step 2.3: Data-base creation:

For the acceleration profiles, the tuned variables are a_{mi1}, a_{mi2}, a_{mi3} and a_{mi4} (where $i = 1, 2$) to get various average velocities of the capsubot. For simplicity, we design $a_{mi2} = a_{mi3}$ and $a_{mi1} = a_{mi4}$ and a fixed value for $a_{mi1} = a_{mi4}$ (maintaining $|a_{mi1}| = |a_{mi4}| < min(\frac{|f_{MM}|}{2m_i}, \frac{|M_{fM}|}{2m_i l_i})$). Only $a_{mi2} = a_{mi3}$ are tuned to get various average velocities of the capsubot. It is noted that if we take $a_{miu2} \neq a_{miu3}$ and $a_{miu1} \neq a_{mi4}$ the data-base size will be larger.

A parameter-set includes a_{mi1}, a_{mi2}, a_{mi3} and a_{mi4} (where $i = 1, 2$). Total number of acceleration profile parameter-sets for linear motion (n_l) and rotational motion (n_r) are:

$$n_l = floor(\frac{|a_{mmax(l)}| - |a_{mmin(l)}|}{a_{mdiff(l)}}) + 1; \quad n_r = floor(\frac{|a_{mmax(r)}| - |a_{mmin(r)}|}{a_{mdiff(r)}}) + 1 \tag{17}$$

where $floor(A)$ rounds the elements of A to the nearest integers less than or equal to A; l and r refers to linear and rotational; $|a_{mmax(l)}|$ and $|a_{mmax(r)}|$ are maximum accelerations, $|a_{mmin(l)}|$ and $|a_{mmin(r)}|$ are minimum accelerations, $a_{mdiff(l)}$ and $a_{mdiff(r)}$ are differences between accelerations of two consecutive profile parameter-sets.

Average linear and angular velocities of the capsubot for all possible profile parameter-sets are calculated using (11) and (14) respectively and stored in the data-base.

4.2.4. Step 2.4: Selection of Profile Parameters

The piece is decided from the desired trajectory by satisfying the constraint (16). In each piece the capsubot needs to follow a behaviour-set from Fig. 3. The desired velocity, $\bar{\dot{x}}_{Md(j)}$ and desired angular velocity, $\bar{\dot{\phi}}_{Md(j)}$ can be calculated as:

$$x_{Md(j)} = \sqrt{(x_j - x_{j-1})^2 - (y_j - y_{j-1})^2}; \quad \phi_{Md(j)} = \tan^{-1}\frac{y_j - y_{j-1}}{x_j - x_{j-1}} \quad \forall \ j = 1, 2, ..n \tag{18}$$

$$\bar{\dot{x}}_{Md(j)} = \frac{x_{Md(j)}}{\frac{T}{2} - t_s}; \quad \bar{\dot{\phi}}_{Md(j)} = \frac{\phi_{Md(j)} - \phi_{Md(j-1)}}{\frac{T}{2} - t_s} \quad \forall \ j = 1, 2, ..n \tag{19}$$

where n is the number of pieces; (x_0, y_0) and $\phi_{Md(0)}$ are the initial capsubot position and orientation.

For A and B (Fig. 3), $|\bar{\dot{x}}_{Md(i)}| \leq \varepsilon_2$ (ε_2 is a small number), thus $\bar{\dot{\phi}}_{Md(j)}$ is modified as:

$$\bar{\dot{\phi}}_{Md(j)} = \frac{\phi_{Md(j)} - \phi_{Md(j-1)}}{T - t_s} \quad \forall \ j = 1, 2, ..n \tag{20}$$

For C and D (Fig. 3), $|\bar{\dot{\phi}}_{Md(j)}| \leq \varepsilon_1$ (ε_1 is a small number), thus $\bar{\dot{x}}_{Md(j)}$ is modified as:

$$\bar{\dot{x}}_{Md(j)} = \frac{x_{Md(j)}}{T - t_s} \quad \forall \ j = 1, 2, ..n \tag{21}$$

Selection:

If $\bar{\dot{\phi}}_{Md}$ is negative (CW rotational motion) IM_2 follows the profile of Fig. 2(a) whereas IM_1 follows $\ddot{x}_{m1d} = -\ddot{x}_{m2d}$. If $\bar{\dot{\phi}}_{Md}$ is positive (CW rotational motion) IM_1 follows the profile of Fig. 2(a) whereas IM_2 follows $\ddot{x}_{m2d} = -\ddot{x}_{m1d}$. Now we need to select the profile parameters (a_{mi1}, a_{mi2}, a_{mi3} and a_{mi4}) which will generate the required desired $\bar{\dot{\phi}}_{Md}$. From the data-base created in section 4.2.3 we get $\bar{\dot{\phi}}_M(q)$, $q = 1, 2, ..n_r$ i.e. all the possible profile parameter-sets and corresponding average angular velocities. Thus the minimum error $\dot{\phi}_{diff}$ can be obtained by (22). The profile parameter-set corresponding to minimum error in (22) is selected.

$$\dot{\phi}_{diff} = min((|\bar{\dot{\phi}}_{Md}| - |\bar{\dot{\phi}}_M(1)|), (|\bar{\dot{\phi}}_{Md}| - |\bar{\dot{\phi}}_M(2)|), ..., (|\bar{\dot{\phi}}_{Md}| - |\bar{\dot{\phi}}_M(n_r)|); \tag{22}$$

If $\bar{\dot{x}}_{Md}$ is positive (FW linear motion) both IMs follow the profile of Fig. 2(a). If $\bar{\dot{x}}_{Md}$ is negative (BW linear motion) IMs follow accelerations with the equal magnitude as in Fig. 2(a) but opposite in direction. From the data-base we get $\bar{\dot{x}}_M(p)$, $p = 1, 2, ..n_l$ i.e. all the possible profile parameter-sets and corresponding average linear velocities. Thus the minimum error $\dot{x}_{diff}$ can be obtained by (23). The profile parameter-set corresponding to minimum error in (23) is selected.

$$\dot{x}_{diff} = min((|\bar{\dot{x}}_{Md}| - |\bar{\dot{x}}_M(1)|), (|\bar{\dot{x}}_{Md}| - |\bar{\dot{x}}_M(2)|), ..., (|\bar{\dot{x}}_{Md}| - |\bar{\dot{x}}_M(n_l)|) \tag{23}$$

4.2.5. Step 2.5: Tuning the Piece Time

The piece time is tuned based on the selected profile parameter-sets. The selected parameter set can only be used for a multiple of cycle time i.e. one cycle or two cycles or three cycles or so on - cycle time is the time to complete all the steps (four steps for linear and rotational mode and, two steps for switching mode) of the acceleration profile. To satisfy this constraint the piece-time T is tuned as follows:

$$Rotation: \ T_{R(tuned)} = t_{r(sel)} \times floor(\frac{T - t_s}{t_{r(sel)}}); \quad Linear: \ T_{l(tuned)} = t_{L(sel)} \times floor(\frac{T - t_s}{t_{l(sel)}}) \tag{24}$$

$$For\ A\ and\ B: \ T_{tuned} = T_{r(tuned)} + t_s; \quad For\ C\ and\ D: \ T_{tuned} = T_{l(tuned)} + t_s \tag{25}$$

$$For\ F,\ G,\ H\ and\ I: \ T_{tuned} = T_{l(tuned)} + T_{r(tuned)} + 2t_s \tag{26}$$

where $t_{l(sel)}$ and $t_{r(sel)}$ are the cycle times of the selected acceleration profiles for linear and rotational motions respectively and t_s is the cycle time for switching mode.

4.2.6. Step 2.6: Modification of the desired angular and linear velocities

We modify the desired angular and linear velocities iteratively using the error in each piece. The expected position of the capsubot after completing each piece is calculated based on the selected profile parameters and tuned piece-time. This position is used to modify the desired angular and linear velocities for the next piece. Thus the modified desired velocities $\bar{\dot{x}}_{Md(j)}$ and $\bar{\dot{\phi}}_{Md(j)}$ are:

$$x_{Md(j)} = \sqrt{(x_j - x_{c(j-1)})^2 - (y_j - y_{c(j-1)})^2}; \quad \phi_{Md(j)} = \tan^{-1}\frac{y_j - y_{c(j-1)}}{x_j - x_{c(j-1)}} \quad \forall \ j = 2, 3, ..n \tag{27}$$

$$\bar{\dot{x}}_{Md(j)} = \frac{x_{Md(j)}}{\frac{T}{2} - t_s}; \quad \bar{\dot{\phi}}_{Md(j)} = \frac{\phi_{Md(j)} - \phi_{Md(j-1)}}{\frac{T}{2} - t_s} \quad \forall \ j = 2, 3, ..n \tag{28}$$

where $(x_{c(j-1)}, y_{c(j-1)})$ is the current position of the capsubot before the tracking of the jth piece of trajectory and can be calculated iteratively as:

$$x_{c(j-1)} = x_{c(j-2)} + s_{j-1}cos(\theta_{j-1}); \quad y_{c(j-1)} = y_{c(j-2)} + s_{j-1}sin(\theta_{j-1}) \tag{29}$$

where θ_{j-1} is the current orientation with respect to O before tracking starts at the jth piece; s_{j-1} is the displacement of the capsubot at the (j-1)th piece.

$$\theta_{j-1} = \theta_{j-2} + T_{R(tuned)}\bar{\dot{\phi}}_{Mu(j-1)sel}; \quad s_{j-1} = T_{R(tuned)}\bar{\dot{x}}_{Mu(j-1)sel} \tag{30}$$

where $\bar{\dot{\phi}}_{Mu(j-1)sel}$ and $\bar{\dot{x}}_{Mu(j-1)sel}$ are the capsubot angular and linear average velocities respectively of the (j-1)th piece for the selected parameters.

4.2.7. Step 2.7: Rules for Implementing Behaviours

We develop the following rules to implement each behaviours:

FW linear motion: (1) Select acceleration profile parameters, (2) Calculate the corresponding $T_{L(tuned)}$, (3) Execute the IMs movement till $T_{elapsed} \leq T_{L(tuned)}$, (4) Set P=1.

BW linear motion: (1) Select acceleration profile parameters, (2) Calculate the corresponding $T_{L(tuned)}$, (3) Execute the IMs movement till $T_{elapsed} \leq T_{L(tuned)}$, (4) Set P=2.

CW rotational motion: (1) Select acceleration profile parameters, (2) Calculate the corresponding $T_{R(tuned)}$, (3) Execute the IMs movement till $T_{elapsed} \leq T_{R(tuned)}$, (4) Set P=3.

CCW rotational motion: (1) Select acceleration profile parameters, (2) Calculate the corresponding $T_{R(tuned)}$, (3) Execute the IMs movement till $T_{elapsed} \leq T_{R(tuned)}$, (4) Set P=4.

Switching to FW linear motion: (1) Decide on the last behaviour - P=1, 2, 3 or 4 means the previous behaviour executed is FW linear, BW linear, CCW rotational or CW rotational respectively, (2.1) If P=1 then $\ddot{x}_{m1d} = \ddot{x}_{m2d} = 0$ for IM_1 and IM_2, (2.2) Elseif P=2 then switching mode ($a_{ms11} = -a_{ms12}$; $a_{ms21} = -a_{ms22}$ and $\ddot{x}_{m1d} = \ddot{x}_{m2d}$) for both the IMs, (2.3) Elseif P=3 then switching

mode ($a_{ms11} = -a_{ms12}$) for IM_1 and $\ddot{x}_{m2d} = 0$ for IM_2, (2.4) Elseif P=4 then switching mode ($a_{ms21} = -a_{ms22}$) for IM_2 and $\ddot{x}_{m1d} = 0$ for IM_1, (3) Execute one switching cycle with selected parameters.

Switching to BW linear motion: (1) Decide on the last behaviour, (2.1) If P=1 then switching mode ($a_{ms11} = -a_{ms12}$; $a_{ms21} = -a_{ms22}$ and $\ddot{x}_{m1d} = \ddot{x}_{m2d}$) for both the IMs , (2.2) Elseif P=2 then $\ddot{x}_{m1d} = \ddot{x}_{m2d} = 0$ for IM_1 and IM_2 , (2.3) Elseif P=3 then switching mode ($a_{ms21} = -a_{ms22}$) for IM_2 and $\ddot{x}_{m1d} = 0$ for IM_1 , (2.4) Elseif P=4 then switching mode ($a_{ms11} = -a_{ms12}$) for IM_1 and $\ddot{x}_{m2d} = 0$ for IM_2, (3) Execute one switching cycle with selected parameters.

Switching to CW rotational motion: (1) Decide on the last behaviour, (2.1) If P=1 then switching mode ($a_{ms21} = -a_{ms22}$) for IM_2 and $\ddot{x}_{m1d} = 0$ for IM_1, (2.2) Elseif P=2 then switching mode ($a_{ms11} = -a_{ms12}$) for IM_1 and $\ddot{x}_{m2d} = 0$ for IM_2, (2.3) Elseif P=3 then switching mode ($a_{ms11} = -a_{ms12}$; $a_{ms21} = -a_{ms22}$ and $\ddot{x}_{m1d} = -\ddot{x}_{m2d}$) for both the IMs, (2.4) Elseif P=4 then $\ddot{x}_{m1d} = \ddot{x}_{m2d} = 0$ for IM_1 and IM_2, (3) Execute one switching cycle with selected parameters.

Switching to CCW rotational motion: (1) Decide on the last behaviour, (2.1) If P=1 then switching mode ($a_{ms11} = -a_{ms12}$) for IM_1 and $\ddot{x}_{m2} = 0$ for IM_2, (2.2) Elseif P=2 then switching mode ($a_{ms21} = -a_{ms22}$) for IM_2 and $\ddot{x}_{m1d} = 0$ for IM_1, (2.3) Elseif P=3 then $\ddot{x}_{m1d} = \ddot{x}_{m2d} = 0$ for IM_1 and IM_2, (2.4) Elseif P=4 then switching mode ($a_{ms11} = -a_{ms12}$; $a_{ms21} = -a_{ms22}$ and $\ddot{x}_{m1d} = -\ddot{x}_{m2d}$) for both the IMs, (3) Execute one switching cycle with selected parameters.

Stationary: (1) Wait for one piece time.

4.3. Step 3: Low-level Control of the IMs

The open loop control laws of the IMs are:

$$F_{m_{id}} = m_i \ddot{x}_{mid} + sgn(\dot{x}_{mid} - \dot{r}_d)\mu_i m_i g \quad \forall \ i = 1, 2 \tag{31}$$

where $\dot{r}_d = \dot{x}_{Md}$ for linear motion mode and, $\dot{r}_d = 0$ for switching mode and rotational motion mode.

The closed loop control law can be selected, using partial feedback linearization [16]

$$F_{m_{id}} = \alpha_i \tau_{id} + \beta_i \tag{32}$$

where $\alpha_i = m_i, \beta_i = sgn(\dot{x}_{mid} - \dot{r}_d)\mu_i m_i g$.

Let $\tilde{x}_i = x_{mi} - x_{mid}$ be the tracking error; choosing the linear control law $\tau_{id} = \ddot{x}_{mid} - k_1\dot{\tilde{x}}_i - k_2\tilde{x}_i$ and applying the control law (32) to (1) we get,

$$\ddot{\tilde{x}}_i + k_{i1}\dot{\tilde{x}}_i + k_{i2}\tilde{x}_i = 0 \tag{33}$$

The values of k_{i1} and k_{i2} can properly be selected using the standard linear control theory. Then by using the control laws (32) the IMs can be made to track the desired trajectories.

5. Simulation Results and Discussion

The parameters used for simulation are taken from the prototype implemented in [15]: $m_1 = m_2 = 6.4gm, \mu_1 = \mu_2 = 0.2, \mu_r = 0.08, \mu = 0.28, F_{m1(max)} = F_{m2(max)} = 1.03N, l_1 = l_2 = 11.5mm, M = 42.9gm, g = 9.8ms^{-2}, w = 7cm, l = 8.7cm$ and $k = 6mm$. The acceleration profile parameters used to create the data-set for the trajectory tracking algorithm are: Linear: $|a_{mmax(l)}| = 20ms^{-2}, |a_{mmin(l)} = 10ms^{-2}|, a_{mdiff(l)} = 0.05ms^{-2}, |a_{mi1}| = |a_{mi4}| = 7ms^{-2}$; Rotation: $|a_{mmax(r)}| = 20ms^{-2}$, $|a_{mmin(r)}| = 9ms^{-2}$, $a_{mdiff(r)} = 0.1ms^{-2}$ and $|a_{mi1}| = |a_{mi4}| = 7ms^{-2}$. The minimum piece-time T calculated from the constraint (16) and the above parameters is 0.4736 s. We used T = 1 s, 2 s and 4 s in simulation to evaluate the impact of T to the control performance. The initial position and orientation of the capsubot are assumed to be (0,0) and 1 rad respectively.

Figures. 5 to 6 show the simulation results for trajectory tracking using the proposed approach for T = 2 s. Figure 7 provides a comparison of the errors in trajectory tracking using various piece-times. From Figs. 5(c) and 5(d), we see the impact of piece-wise tracking in the simulated translation and steering angle. In every piece the capsubot first corrects the orientation and then it travel the line joining the start and end point of the piece. Thus we see in Fig. 5(c) that the translation graph remains flat at the beginning of the piece and then increases whereas the steering angle graph changes for the first portion of the piece and then remains flat for the rest of the piece. The error in the simulated trajectory remains within certain limit. The steering angle tracking error is within -0.2879 rad to 0.2984 rad. The mean absolute error of steering angle tracking is 0.0744 rad. We are able to get this result by modifying the desired angular and linear velocities in each piece. The error can further be reduced by decreasing the piece time. We get the simulated trajectories of Fig. 6 by using the following equations:

$$x_j = x_{j-1} + (d_j - d_{j-1})cos(\phi_j); \quad y_j = y_{j-1} + (d_j - d_{j-1})sin(\phi_j) \tag{34}$$

It is noted from Fig. 6 that the capsubot follows the trajectory quite accurately. From Figs. 6(a) and 7(b) (the curve for T=2s) we see that the error increases at the beginning of each piece and then goes close to zero at the end of the piece. The capsubot corrects its steering angle in the first portion of the piece when it does not have any translation. In the second portion the capsubot performs translation and thus the error in x decreases. Figure 6(b) compares the desired and simulated y and shows the error in trajectory tracking. From Figs. 6(a) and 6(b) we see that the error patterns are same for both x and y tracking. The error increases at the beginning of each piece and then decreases close to zero at the end of the piece for the same reason as x tracking. The error range for x tracking is -0.1751 cm to 0.4361 cm and the error range for y tracking is -0.7261 cm to 0.6736 cm. The mean absolute error of x position is 0.1628 cm whereas mean absolute error of y position is 0.2328 cm.

From Fig. 6(c), it is noted that the capsubot can follow the desired trajectory with a small error. However the simulated trajectory is not smooth as the capsubot follows the trajectory in a piece-wise manner. By choosing a smaller piece-time the simulated trajectory can be made smoother. This simulation demonstrates the feasibility of the piece-wise trajectory tracking algorithm for this class of underactuated robots. From Fig. 7 and Table 1, we observe that increasing peace-time T will increase the errors in x, y, and steering angle while the computation time will decrease. This is easy to understand from Fig. 2(c).

The uncertainties and disturbances have an impact on the tracking performance of the algorithm. We considered the parameter uncertainty of friction which is one of the dominated uncertainties. Table 2 provides the tracking errors of the algorithm for various friction uncertainties. It is seen from Table 2 that errors increase with increasing uncertainty. From Table 2 we can see that the performance of the algorithm is acceptable as long as the uncertainties remain within -10% to $+10\%$. One way to increase the robustness of the algorithm with respect to friction uncertainty is to model friction dynamically in each piece and then use it in the next piece. In our future research we will explore other control methods such as adaptive control, robust control etc. to improve the sensitivity and robustness of the tracking algorithm to model uncertainties and disturbances.

Table 1: Comparison of the algorithm performance with various piece-times

T (s)	Maximum absolute error			Mean absolute error			Computation
	x (cm)	y (cm)	ϕ(rad)	x (cm)	y (cm)	ϕ(rad)	Time (ms)
1	0.2722	0.4072	0.2344	0.0928	0.1340	0.0497	67
2	0.4361	0.7261	0.2984	0.1628	0.2328	0.0744	59
4	1.0900	1.3125	0.4594	0.4041	0.3733	0.1089	55

6. Conclusions and Future Works

This paper proposed a trajectory tracking algorithm combining piece-wise and behaviour-based control to solve the trajectory tracking problem of an underactuated 2D capsubot - the capsubot belongs to a class of underactuated systems which work on the principle of internal reaction force. Based on the authors' knowledge, it is the first time the trajectory tracking for this class of underactuated systems has been addressed. We defined the basis behaviours and formed required behaviour-sets to track the trajectory. The selection algorithm chooses the appropriate behaviour-set to track each piece of the trajectory. The rules were used to execute individual behaviours of the selected behaviour-set. Partial feedback linearization control was used for low level IMs' motion control. Simulation was

Table 2: Comparison of the algorithm performance for various friction changes where piece time = 1s

Friction variation (%)		Maximum absolute error			Mean absolute error			Relative mean absolute error* (%)		
Transla-tional (μ)	Rota-tional (μ_r)	x (cm)	y (cm)	ϕ (rad)	x (cm)	y (cm)	ϕ (rad)	x	y	ϕ
0	0	0.272	0.407	0.234	0.093	0.134	0.05	0.4802	3.423	4.540
-10 to 10	0	0.418	0.604	0.234	0.152	0.18	0.05	0.790	4.587	4.540
0	-10 to 10	0.959	0.772	0.230	0.325	0.30	0.053	1.682	7.643	4.864
-10 to 10	-10 to 10	0.834	0.944	0.230	0.319	0.377	0.053	1.650	9.626	4.864
-15 to 15	0	0.419	0.732	0.234	0.132	0.258	0.050	0.684	6.595	4.540
0	-15 to 15	1.891	1.193	0.237	0.769	0.557	0.065	3.977	14.24	5.928
-15 to 15	-15 to 15	1.938	1.610	0.237	0.844	0.738	0.065	4.363	18.84	5.928

* Relative mean absolute error = (mean absolute error / mean absolute desired value)*100 %

performed using the parameters of the capsubot prototype of [15]. The simulation results shown the feasibility of the proposed algorithms and rules.

We will further analyse the robustness of the control method to modelling uncertainties and/or disturbances in our future research. We aim to implement the proposed trajectory tracking control approach in a capsubot prototype in future. Ultimately this research will lead to the wireless control of the robot within in-vivo and industrial environments.

References

[1] Liu, Y., Yu, H.: 'A survey of underactuated mechanical systems', IET Control Theory & Applications, 2013, 7, (7), pp. 1–7

[2] Lopez-Martnez, M., Acosta, J., Cano, J.: 'Non-linear sliding mode surfaces for a class of underactuated mechanical systems', IET Control Theory & Applications, 2010, 4, (10), pp. 2195-2204.

[3] H. Yu, H., M. Huda, M.N., S. Wane, S.: 'A novel acceleration profile for the motion control of capsubots', Proc. 2011 IEEE Int. Conf. on Robotics and Automation, 2011, pp. 2437-2442.

[4] Valdastri, P., Webster, R., Quaglia, C., et al.: 'A new mechanism for mesoscale legged locomotion in compliant tubular environments', IEEE Transactions on Robotics, 2009, 25, (5), pp. 1047-1057.

[5] Yim, S., Sitti, M.: 'Design and rolling locomotion of a magnetically actuated soft capsule endoscope', IEEE Transactions on Robotics, 2012, 28, (1), pp. 183-194

[6] Carpi, F., Kastelein, N., Talcott, M., et al.: 'Magnetically controllable gastrointestinal steering of video capsules', IEEE Transactions on Biomedical Engineering, 2011, 58, (2), pp. 231-234

[7] Carta, R., Sfakiotakis, M., Pateromichelakis, N. et al.: 'A multi-coil inductive powering system for an endoscopic capsule with vibratory actuation', Sensors and Actuators A: Physical, 2011, 172, (1), pp. 253-258

[8] Zhang, Y., Jiang, S., Zhang, X., et al.: 'A variable-diameter capsule robot based on multiple wedge effects', IEEE/ASME Transactions on Mechatronics, 2011, 16, (2), pp. 241-254

[9] Lee, N., Kamamichi, N., Li, H., et al.: 'Control system design and experimental verification of capsubot', Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, 2008, pp. 1927-1932

[10] Liu, Y., Yu, H., Yang, Y.: 'Analysis and Control of a Capsubot', Proc. of the 17th World Congress of the International Federation of Automatic Control, July 2008, pp. 756-761.

[11] Chernousko, F.: 'The optimum rectilinear motion of a two-mass system', Journal of applied Mathematics and Mechanics, 2002, 66, (1), pp. 1-7

[12] Brooks, R.: 'A robust layered control system for a mobile robot', IEEE Journal of Robotics and Automation, 1986, 2, (1), pp. 14-23.

[13] Michaud, F.: 'Behavior-based systems', in Siciliano, B., Khatib, O. (Eds.): Springer handbook of robotics (Springer Press, 2008, 1st edn.), pp. 891-909

[14] Proetzsch, M., Luksch, T., Berns, K.: 'Development of complex robotic systems using the behavior-based control architecture iB2C', Robotics and Autonomous Systems, 2010, 58, (1), pp. 46-67.

[15] Huda, M.N., Yu, H., Goodwin, M.J.,: 'Experimental study of a capsubot for two dimensional movements', Proc. 2012 UKACC International Conference on Control. IEEE, 2012, pp. 108-113 (Best student paper)

[16] Yu, H., Liu, Y., Yang, T.: 'Closed-loop tracking control of a pendulum-driven cart-pole underactuated system', Proc. of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering, 2008, 222, (2), pp. 109-125