

無線感測網路中以預測區間涵蓋為優先排程之多目標追蹤

黃士庭、陳震宇

國立東華大學電機工程學系

摘要 — 在無線感測網路的多目標追蹤研究上，其主要問題是如何提高追蹤精準度和降低能量消耗。本篇論文提出利用預測方法預測目標狀態，根據預測目標狀態和感測節點佈署之間的關係挑選檢測區間。且依照所挑選的檢測區間，來喚醒並形成叢集來追蹤目標。在多目標追蹤上，我們將利用調整檢測區間，以時間來區隔衝突節點所衝突的時間，以達到多目標的成功追蹤。最後模擬結果表明，提出的方法確實能夠提高追蹤成功的機率。

一、簡介

無線感測網路是由大量的感測節點部屬在特定區域所組成，每個感測節點的配備有嵌入式處理器、感測元件、儲存和無線通訊設備。可被廣泛的應用到各個地方，如：環境的監控、軍事應用、健康應用、家庭應用、商業管理和目標追蹤等[1][2]。其架構主要可分為匯集點 (Sink node) 和感測節點 (Sensor node)。匯集點主要是收集感測節點監控環境所得到的資料所；感測節點主要是收集自己監控範圍 (Sensing range) 內的環境資訊。

在無線感測網路多種監測環境應用中，目標追蹤 (Target Tracking) 一直是重要的應用，如軍事上應用偵測敵軍動向、追蹤動物移動路徑、家庭看護等。然而追蹤會移動的目標比偵測固定的區域要困難得多，為了得到目標的即時位置資訊，目標的位置資訊必須隨時更新，因此感測節點需要隨時都能對目標進行定位工作，且還要不停的監控目標移動方式，以降低丟失目標機會，並將目標資訊提供給使用者。

以下分各個章節依序介紹本論文，在第二章將介紹相關研究，第三章將會說明我們所提出的多目標追蹤演算法，第四章則是模擬與實驗，最後提出我們的結論。

二、相關研究

在本篇論文主要是使用預測方法預測目標位置，根據預測目標位置喚醒感測節點形成叢集來追蹤目標為 Prediction Based Clustering，並將針對叢集式 (Cluster Based) 的目標追蹤來進行研究。執行追蹤任務時，數個感測節點將形成叢集，並會從中挑選出叢集首 (Cluster Head, CH) 其餘為叢集成員 (Cluster Member, CM)。叢集成員量測目標資訊，將此資訊傳給叢集首，叢集首將收集資料、計算和預測目標資訊，決定下一時間該執行追蹤任務的感測節點，所以叢集首是較重要且耗能的。上述所敘是一個動態的叢集式 (Dynamic Clustering) 架構 [3][4]。

[5][6][7]提出自適的挑選檢測區間 (Sampling

interval)。[6]提出一種分佈式自適的多個感測節點調度，與感測節點的合作執行目標追蹤在無線感測網路。基於 Extended Kalman filter (EKF) 預測目標位置的不確定性，去判斷位置的不確定性的結果是否滿足門檻值，去選擇出一個最佳的檢測區間。在選擇感測節點上，假設感測節點檢測出目標的機率並且設定機率門檻值，並根據感測節點的聯合感測機率以達門檻機率來選擇 n 個感測節點組成叢集來執行追蹤目標任務，並且從中挑選出叢集首來收集成員訊息，接著計算下一步的追蹤任務，做單一目標的追蹤。

[7]假設感測節點同一時間只能追蹤一個目標的情況下所提出的多目標追蹤演算法 (DMTT)，使用目標歷史軌跡來計算檢測區間，當目標直線移動檢測區間將會變大，不規則移動將會變小。在選擇節點上，使用 EKF 來預測目標位置，並基於擴展型卡爾曼濾波器中的 Predicted covariance matrix 來計算感測節點與預測目標位置的 Mahalanobis distance，從中挑選出 Mahalanobis distance 最短的 n 個感測節點成叢集執行追蹤任務。在多目標追蹤上有發生衝突節點時，衝突感測節點會依據自己位置與預測目標的 Mahalanobis distance 和目標重要性之間的權重值去選擇要追蹤哪個目標，並會選擇另外的感測節點去追蹤另一個目標。

在[5][6][7]雖然可根據目標移動狀態或根據追蹤誤差門檻值選擇檢測區間，但是並沒考量節點佈署的問題，當選擇出檢測區間，但附近感測節點不足，將會造成目標丟失或者定位誤差過大等問題。

[7]在多目標追蹤，由於感測節點硬體的受限上，假設目標在一次追蹤任務只能追蹤一個目標，所以將會有衝突節點發生的問題。但是如果在衝突節點發生時，只利用文獻所提出解決衝突節點 Distributed multi-target selection (DMS) 的方法，可能在節點分佈較少的地方會有無節點可換的問題。

三、預測演算法

[8]基於機率目標預測 (Prediction Base Target Perdition, PTP) 演算法利用三個步驟來預測目標下一步可能出現的範圍：當前狀態計算、基於運動學的預測和基於機率的預測。

3.1 當前狀態計

計算在時間點 $t(k)$ 的目標當前狀態向量為：

$$X(k) = (t(k), x(k), y(k), \vec{v}(k), \vec{a}(k)) \quad (1)$$

$L(k) = (x(k), y(k))$ 為目標物座標上之位置， $\vec{v}(k)$ 與

$\bar{a}(k)$ 為目標物在時間區段 $(t(k-1), t(k))$ 內的平均速度向量以及平均加速度向量。

3.2 基於運動學的預測

假設加速度 $\bar{a}(k)$ 在時間 $t(k)$ 到 $t(k+1)$ 中皆保持不變，因為加速度是二階導數的位移，其變化率(即：加加速度)為三階導數的位移，這在泰勒多項式中通常是被忽略的。

$$\begin{aligned}\bar{\hat{a}}(k+1) &= \bar{a}(k) \\ \bar{\hat{v}}(k+1) &= \bar{v}(k) + \bar{\hat{a}}(k+1) \cdot \Delta t \\ \bar{\hat{S}}(k+1) &= \bar{v}(k+1) \cdot \Delta t + \frac{1}{2} \cdot \bar{\hat{a}}(k+1) \cdot \Delta t^2 \\ \hat{L}(k+1) &= L(k) + \bar{\hat{S}}(k+1)\end{aligned}\quad (2)$$

其中， $\bar{\hat{S}}(k+1)$ 為時間 $t(k)$ 到 $t(k+1)$ 中目標的預測移動向量， $\hat{L}(k+1)$ 則為目標在時間點 $t(k+1)$ 時的預測位置。

3.3 基於機率的預測

設置一個機率模型， $S(k+1)$ 和 $\Delta(k+1)$ 為隨機變數， $\bar{\hat{S}}(k+1) = (S(k+1), \Delta(k+1))$ 。

假設 $S(k+1)$ 為高斯， $S(k+1) \sim N(\mu_{S(k+1)}, \sigma_{S(k+1)}^2)$ 。加速度保持不變，其平均值計算為：

$$\mu_{S(k+1)} = \left\| \bar{\hat{v}}(k+1) \cdot \Delta t + \frac{1}{2} \cdot \bar{\hat{a}}(k+1) \cdot \Delta t^2 \right\| \quad (3)$$

而 $\sigma_{S(k+1)}^2$ 基於高斯分布「68-95-99.7 rule」，在 $S(k+1)$ 可能會發生的改變為 $\|\bar{v}(k) \cdot \Delta t\|$ 和 $\|\bar{\hat{v}}(k+1) \cdot \Delta t\|$ ，在此範圍的的機率近似為 68%。因此設置位移的機率模型為：

$$\sigma_{S(k+1)}^2 = \left(\left\| \bar{\hat{S}}(k+1) \right\| - \left\| \bar{\hat{v}}(k+1) \cdot \Delta t \right\| \right)^2 \quad (4)$$

由於我們檢測區間有時間上的限制，所以我們在此目標短時間的移動根據慣性可以視為線性移動，故在此我們將不考慮角度。

四、多目標追蹤架構

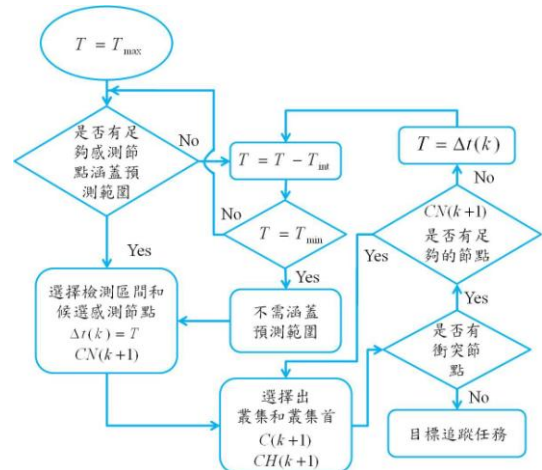
假設要監測一個特定的區域為二維空間，將感測節點隨機均勻部屬在這特定區域裡進行目標追蹤的任務。每個感測節點的硬體都相同，有相同的通訊範圍為半徑 R 和感測範圍半徑為 r 。所有的感測節點可透過 Global Positioning System (GPS) 或一些定位技術來得知自己的座標位置，感測節點會知道自己和自己鄰居資訊，資訊包含感測節點編號 (ID)、座標和能量。感測節點時間為同步。

為了節省能量消耗，感測節點使用低消耗的通訊通道並且會讓特定區域內部還不需要用到的感測節點進入睡

眠模式 (Sleep Mode)，在特定區域的邊境感測節點均為感測模式和通訊模式。因此，在追蹤目標時如果需要這些內部感測節點追蹤目標，那這些節點將會被「喚醒」。特定節點會形成叢集執行目標追蹤任務，當任務結束時進入睡眠模式。為了簡化，測量的目標的不確定性 (uncertainty) 和感測節點的誤報 (false alarms) 在這裡不考慮。

我們所提出的以預測區間涵蓋為優先排程之多目標追蹤演算法 (PCF) 主要可分為下列三個步驟：

1. 基於預測目標狀態和感測節點的位置環境來決定來選擇合適的檢測區間和可使用的候選節點 (Candidate node, CN)
2. 從候選節點選擇 n 個感測節點形成叢集，並從中選擇出叢集首，其他則為其叢集成員。叢集首會收集成員資訊，接著計算出下一步的任務。
3. 追蹤多目標產生衝突節點情況時，替另一個目標挑選其他合適感測節點或檢測區間，進而重新分配其追蹤感測節點。



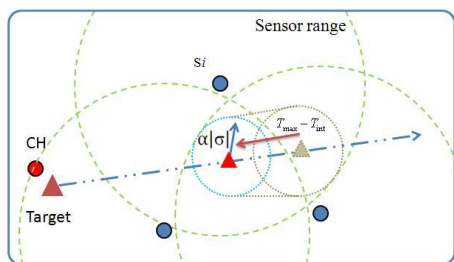
圖一：多目標追蹤演算法完整流程圖

4.1 選擇檢測區間

檢測區間 $(\Delta t(k) = t(k+1) - t(k))$ 為一自適的檢測區間，會根據每段時態以及預測後的資訊加以調整，而不是每次檢測皆使用相同的檢測區間。將檢測區間挑選限制在一個範圍內 $[T_{\min}, T_{\max}]$ ， $T_{\min} \leq \Delta t \leq T_{\max}$ ，而限制檢測區間大小的原因有以下兩點： T_{\min} 為最小檢測區間，最小檢測區間必須去大於感測節點執行追蹤任務所需的時間，追蹤任務包括了：感測時間、傳輸延遲、通道存取時間和資料處理時間，系統必須在當前追蹤任務均執行完畢後才能夠再去進行下一次的檢測。 T_{\max} 為最大檢測區間，最大檢測區間不能過長，因其大多數的目標移動狀態皆為非線性之移動，當檢測區間過長，可能造成預測目標位置和真實目標位置差距過大。

基於 PTP 可預測目標位移可能落在的範圍 σ ，由於沒考慮角度，所以我們以預測目標座標為中心， σ 為半徑，此為預測範圍，並根據感測節點 S_i 的感測半徑 r 可涵蓋之範圍，找出可於下一時態中能執行追蹤任務之節點群，如圖一，判斷可執行追蹤任務節點群之數量，是

否能夠滿足演算法所規定之需求。



圖二：檢測區間選擇示意圖

我們根據上述的計算預測範圍和感測器部屬位置將可寫出判斷式：

$$\hat{D}(k+1|k, s_i) + \alpha|\sigma| \leq r \quad (5)$$

其中 $\hat{D}(k+1|k, s_i)$ 為預測目標座標 $\hat{L}(k+1)$ 和感測節點 L_{s_i} 之間的距離、 σ 為其標準差、 r 則為感測節點之感測半徑、 α 為可設定的調變參數。利用此判斷式可計算出滿足 n 個感測節點的檢測區間。演算法將由最大檢測區間 $T = T_{\max}$ 開始並逐步降低 T_{\min} ，直到可執行追蹤任務之節點群滿足 n 個節點數量。如果無法挑選到滿足 n 個感測節點的檢測區間，將以 T_{\min} 作為其檢測區間，並由 T_{\min} 來預測目標位置 $\hat{L}(k+1)$ ，然後去選擇出 n 個最近的感測節點，此時不須涵蓋預測範圍。

4.2 選擇任務感測節點和叢集首

在每次的目標追蹤任務時，選擇一組感測節點形成叢集來執行追蹤任務，並從叢集 $C(k+1)$ 中選出叢集首，其他則為叢集成員。我們從滿足 n 個感測節點選出的檢測時間區 $\Delta t(k)$ ，假設在此時有 m 個候選節點存在，則有 $CN(k+1) = \{s_1, s_2, \dots, s_m\}$ 個可執行追蹤任務候選感測節點。從 m 個感測節點中挑選出 n 個來執行追蹤任務，這 n 個感測節點將形成一個叢集 $C(k+1) = \{s_1, s_2, \dots, s_n\}$ 。

在挑選執行任務的感測節點時，將優先選擇離預測目標位置最近的感測節點來執行追蹤任務。可防止目標移動時因外在不確定性因素而導致目標移動到我們所選定感測節點的檢測範圍之外。依據挑選的檢測區間 Δt 來預測目標物之位置 $\hat{L}(k+1)$ ，計算出每個感測節點與預測目標位置的歐幾里得距離 $\hat{D}(k+1|k, s_i)$ ，並且從 $CN(k+1)$ 挑選出距離預測目標位置最近的 n 個感測節點，這 n 個感測節點會形成一個叢集 $C(k+1)$ 。

由於叢集首在執行追蹤任務時，要執行的任務較多所以相對地所消耗的能量也較高，因此在選擇叢集首時，將會以感測節點剩餘的能量做考量。而挑選叢集首的方程式如下：

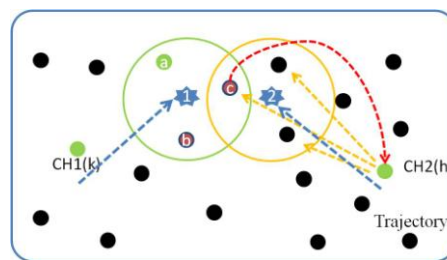
$$CH(k+1) = \arg \max_{s_i} E_{s_i}, \quad s_i \in C(k+1) \quad (5)$$

其中 E_{s_i} 為感測節點的剩餘能量。

4.3 衝突節點分配機制

在多目標追蹤上，我們假設每個節點在執行一次任務時間內只能追蹤一個目標，而之前所提到 T_{\min} 為感測節點執行追蹤任務所需之處理時間，所以我們假設感測節點每次執行追蹤任務時間都為 T_{\min} 。因此當兩個目標以上且均使用同一個感測節點時，其感測節點會在同一個時間內收到兩個需執行的追蹤任務，而此節點我們將稱為衝突節點 (Conflict Node)。

由於叢集首在通知下一次執行追蹤任務的叢集後，將緊接著執行其他任務或者進入睡眠模式，所以我們將設定為較晚傳送任務資訊到衝突節點的叢集首，需要進行排除衝突節點分配機制。



圖三：衝突節點分配機制

如圖三，假設在某一時間點上感測節點 c 收到了兩個相同時間的追蹤任務資訊，此時 c 節點即為衝突節點，衝突節點 c 將會選擇較慢通知的叢集首，並對其回傳一個衝突封包，包含：執行任務的時間、叢集成員資訊等。 $CH2$ 進入衝突節點分配機制，先以當時的檢測區間判斷是否有其他感測節點可使用。如果沒有，將會調整檢測區間，並且會發送衝突封包資訊，利用時間區隔開衝突的時間區段，來達成多目標的追蹤。

衝突節點分配機制分成兩個部分：

1. 從候選節點 $CN(k+1)$ 排除衝突節點，判斷感測節點是否足夠。如果候選節點不夠，將進入 2。如果足夠將會重新進入選擇任務感測節點與叢集首。
2. $CH(k)$ 可得知衝突節點的任務時間，將依照此資訊去調整檢測區間。 $CH(k)$ 從挑選出來的檢測區間，重新進入選擇檢測區間開始計算，檢測區間由 $T = \Delta t(k)$ 開始帶入計算 (此時衝突節點不能加入計算，衝突時間區隔以後才能加入計算)。

五、 模擬實驗

本章以提出的多目標追蹤演算法架構做模擬實驗分析，並與上述所提到的 DMMT 做比較。

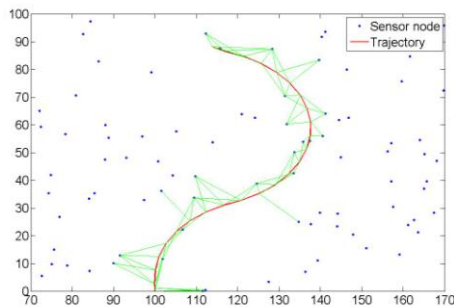
5.1 環境模擬設置

本模擬使用 Matlab，將感測節點隨機均勻佈署在 $100m \times 100m$ 的範圍中，每個感測節點都有一樣的通訊半徑 $R = 30m$ 、感測半徑 $r = 15m$ 。 $T_{\min} = 0.01$ 。檢測區間範圍限制 $[T_{\max} \ T_{\min}] = [0.1 \ 0.5]$ 。在此我們假設每次檢測目標都需要 $n=3$ 個感測節點，目標有在 3 個節點感測範圍內即為成功檢測、定位，給出定位模型為

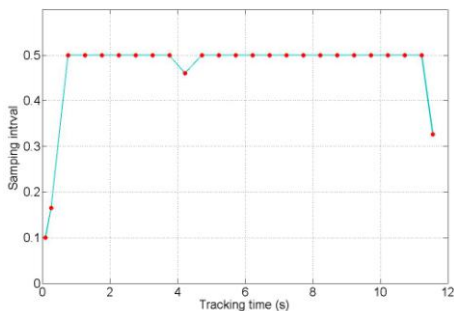
$$\begin{aligned} x(k) &= tx(k) + v_x(k) \\ y(k) &= ty(k) + v_y(k) \end{aligned} \quad (6)$$

$(tx(k) ty(k))$ 為目標的真實座標， $v_x(k)$ 、 $v_y(k)$ 為零均值白高斯量測雜訊，變異數 $\sigma_x^2 = 0.001$ 和 $\sigma_y^2 = 0.001$ 。

圖三，感測節點佈署密度為 $1 (\text{node}/100\text{m}^2)$ ，目標速度為 10m/s S 型路徑，預測範圍的可調變參數 $\alpha = 3$ ，經由圖四，可看出目標的追蹤方式，感測節點隨著目標移動不斷地被喚醒執行追蹤任務。圖四，在感測節點較稀疏的地方，由於節點的感測範圍無法涵蓋預測範圍，而選取較小的檢測區間。



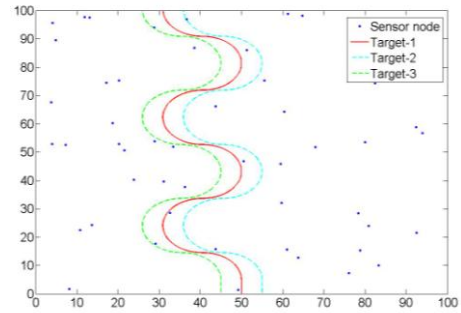
圖三：目標速度為 10m/s 的 S 行路徑



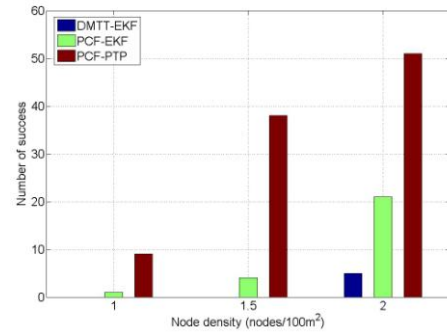
圖四：檢測區間的變化

在此我們假設每次檢測目標都需要 $n=3$ 個感測節點，如果有一次檢測沒有滿足 3 個節點，那視為失敗的追蹤。我們模擬 100 種感測節點的隨機分布的情況，3 種感測節點佈署密度做比較，在 100 次的模擬中目標為相同速度相同路徑移動，改變的只有感測節點佈署位置。

圖六，我們所提出的方法 PCF-PTP 和 DMTT 並且加入 PCF-EKF 做比較。PCF-EKF 使用我們所提出選擇出檢測區間和排除衝突節點方法，預測方法改用 EKF [5] [6] (EKF 無預測範圍所以只需涵蓋預測位置)。EKF 的初始設定，先讓執行 2 次量測目標位置計算出目標速度和座標給定初始值 $X(0)$ ， $P(0) = 10I$ ， I 為 4×4 單位矩陣。 $q = 10$ 。



圖五：三目標速度 10m/s 的 S 型路徑



圖六：三目標速度 10m/s 的 S 型路徑追蹤成功次數

六、 結論

從模擬可看出，我們所提出的方法 PCF，根據感測節點佈署環境，挑選檢測區間可以有較高的追蹤成功機率。在多目標追蹤方面，在感測節點較稀疏的地方，可透過調整檢測區間排除衝突節點，讓被追蹤的多個目標都能在有足夠的感測節點的時間點被檢測，而提高成功追蹤的機率。

參考文獻

- [1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor network: A survey," in *Proc. of Comput. Netw.*, vol. 38, no. 4, pp. 393-442, Mar. 2002.
- [2] T. Malik, M. Sanjay, "Sensor networks: an overview," in *Proc. of Potentials, IEEE*, vol.22, no.2, pp.20,23, April-May 2003..
- [3] M. Fayyaz, "Classification of Object Tracking Techniques in Wireless Sensor Networks" in *Proc. of Wireless Sensor Network*, vol.3, pp. 121-124, 2011.
- [4] K. Ramya, K. Praveen Kumar and Dr. V. Srinivas Rao, "A Survey on Target Tracking Techniques in Wireless Sensor Networks," in *Proc. of IJCSSES Vol.3*, no.4, August 2012.
- [5] J. Lin, W. Xiao, F. L. Lewis, and L. Xie, "Energy-efficient distributed adaptive multisensor scheduling for target tracking in wireless sensor networks," in *Proc. of IEEE Trans. Intrum. Meas.*, vol. 58, no. 6, pp. 1886-1897, Jun. 2009.
- [6] W. Xiao, S. Zhang, J. Lin, "Energy-efficient adaptive sensor scheduling for target tracking in wireless sensor networks," in *Proc. of J. Control Theory Appl.*, p. 86-92, 2010.
- [7] Y. E M. Hamouda and C. Phillips, "Adaptive sampling for energy-efficient collaborative multi-target tracking in wireless sensor networks," in *Proc. of Wireless Sensor Systems, IET*, vol.1, no.1, pp.15-25, March 2011.
- [8] Bo Jiang, B. Ravindran, and Hyeonjoong Cho, "Probability-Based Prediction and Sleep Scheduling for Energy-Efficient Target Tracking in Sensor Networks," in *Proc. of Mobile Computing, IEEE Transactions on*, vol.12, no.4, pp.735-747, April 2013.