

ウェアラブルセンサーを用いた繰り返し作業の サイクルタイム自動計測手法

Unsupervised Method to Automatically Measure Lead Time of Repetitive Processes Using Acceleration Data from Wearable Sensor on Worker

浪岡 保男 NAMIOKA Yasuo 前川 卓也 MAEKAWA Takuya

量産工程において、繰り返し行われる組立・検査作業などに掛かるサイクルタイムは、その工程の生産性を示すための重要な指標の一つである。

東芝は、国立大学法人 大阪大学と共同で、作業者の手首に装着したウェアラブルセンサーから得られる加速度データを用いて、教師なしで作業工程のサイクルタイムを推定する手法を開発した。サイクルタイムを推定するために、時系列の加速度データから、作業ごとに一度現れるセグメント（モチーフ）を抽出する。このとき、事前の情報としては、その工程に設定されている標準作業時間だけを用いる。この繰り返し抽出されるモチーフ間の時間から、作業に要するサイクルタイムを自動計測する。実際の製造現場で、最終組立及び検査工程での検証を行い、この手法の有効性が確認できた。

Lead time, which is the time required to complete an operation in repetitive processes including assembly and inspection processes, is a key parameter in mass production corresponding to the productivity of a factory.

In cooperation with Osaka University, Toshiba has developed a method applying unsupervised method to automatically measure lead time in a mass production line using acceleration data obtained by a wearable sensor on a worker's wrist. This method can automatically extract a motif, which is a distinctive segment that occurs once in each operation period, from time-series acceleration data using only prior knowledge about a predefined standard takt time of the operation process, and consequently estimate the lead time from the interval of occurrence of the motif. We have confirmed the effectiveness of this method through application to final-assembly and inspection operations at our manufacturing sites.

1. まえがき

近年、センサー技術の進展により、ウェアラブルセンサーを用いた行動認識に関する研究が様々な分野で行われており、製造現場でも、作業者の作業分析などに応用されている^{(1), (2)}。東芝は、製造工程の組立作業に着目し、ウェアラブルセンサーを用いて、繰り返し作業におけるサイクルタイムの自動推定技術の開発を進めている^{(3), (4)}。

組立・検査工程は、自動化を進めているが今も多数の作業者が並ぶライン生産方式が多く採用されており、作業効率は、生産性向上の重要なファクターである。組立・検査工程では、製品ごとに定められた作業工程が繰り返し行われており、所定の生産性を得るために、各作業工程には、標準作業時間が定められる。作業者は、一つの製品に対する作業工程を終えると、次工程に製品を渡していくため、一部の作業工程で標準作業時間を超過すると、製造ライン全体の生産性が低下する要因となる。

各作業工程のサイクルタイムをモニタリングすることは、生産性の維持・管理に有効なことから、サイクルタイムを継続的に自動計測するためのアプローチとして、普及が目覚ましいウェアラブルセンサーが活用されている。一般的には、作業工程に含まれる作業完了時の行動を認識し、その行動の出現間隔から所要時間を推定する方法が考えられる。推定には、機械学習の適用が多く提案されている。しかし、自動計測を実現するためには、作業工程に含まれる複数の作業内容ごとにセンサーでデータを取得し、それぞれの作業動作を事前に学習する必要があり、学習コストが大きくなることが課題となっている。更に、同じ作業でも、センサーデータに現れる特徴には個人差があるため、作業者ごとに動作を学習するコストも膨大になる。

そこで、当社は、学習コストが課題となる場合の解決方法として、機械学習やデータマイニングで用いるアルゴリズムを応用し、事前の学習を必要としない、教師なしの作業サイクルタイム自動計測手法を、大阪大学と共同で開発した。こ

ここでは、この手法の概要を述べる。

2. 開発した手法

2.1 開発した手法の概要

開発した手法では、あらかじめ標準作業時間を与えるだけで、センサーデータに繰り返し出現する特徴的なパターンをモチーフとして発見し、モチーフが出現する時間間隔から作業時間を推定する。サイクルタイムは、作業状況の変化などにより変動するが、非線形システムの同定などに使われるパーティクルフィルターを用いることで、モチーフの非線形な出現を追跡できる。パーティクルフィルターは、複数のモチーフの候補をパーティクルとして追跡するが、このときデータ中の類似度の高い部分を発見するために、センサーデータから切り出したセグメントをパーティクルと比較する。

この手法の概要を図1に示す。作業開始直後の t_{init} 間のセンサーデータからランダムに切り出した様々なモチーフを、

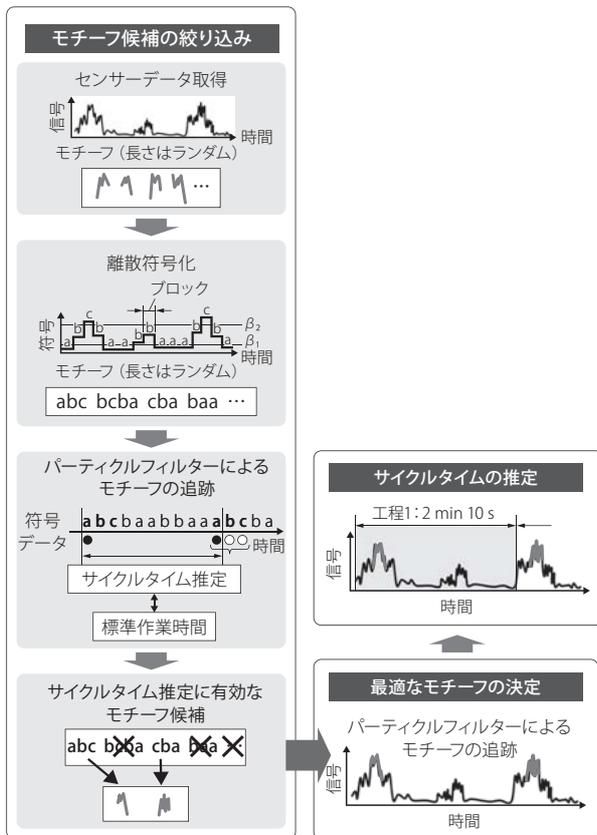


図1. サイクルタイムの自動計測手法の概要

符号化した信号でモチーフ候補を絞り込み、符号化する前の元信号で最適なモチーフを決定することで、サイクルタイムを推定する。

Overview of newly developed method to automatically measure lead time

2.2節で説明する作業モデルに従って評価して採用順位を定める。モチーフの発見では、センサーデータの全系列から切り出された部分区間である各セグメントと注目するモチーフとの類似度を、時系列パターンのマッチング法であるDTW (Dynamic time warping)を用いて計算し、モチーフの出現を追跡する。DTWでは、網羅的に切り出した多数のモチーフに対して類似度の計算を行うため、多大な計算量を必要とする。そこで、この手法では、有用なモチーフの候補を、符号化したセンサーデータを用いて高速に選定する。ここで、有用であると考えられるモチーフとは、各作業工程において一度しか出現しない作業動作中に含まれるパターンのことである。そして、選定された有用なモチーフに対してだけ、DTWを用いて詳細に出現を追跡し、最終的に最も適したモチーフを見付け、最適なモチーフの出現間隔から各作業工程のサイクルタイムを求めめる。

2.2 作業モデル

各作業工程において、あらかじめ定められた標準作業時間を平均とするガウス分布によって表される作業モデルを用いる。また、作業工程に含まれる作業内容が条件によって異なり標準作業時間が変化することがある場合、作業モデルは、それぞれの異なる標準作業時間を平均とするガウス分布の混合で表現される。例えば、作業工程に2 minと9 minという二つの作業内容が含まれる場合、図2で示されるような、2混合の混合ガウス分布を作業モデルとして用いる。

2.3 モチーフ候補の絞り込み

まず、作業開始時点から t_{ms} までの間の作業工程で取得した短時間のセンサーデータだけを用いて、作業モデルに従って繰り返し出現する有用なモチーフを少ない計算コストで絞り込む。開発した手法では、作業開始時点から t_{ms} まで

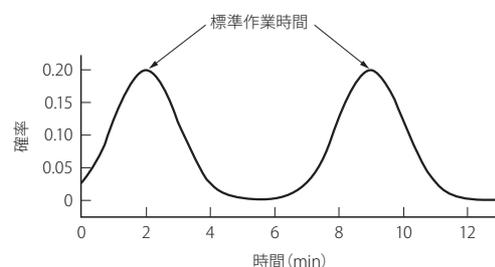


図2. 作業モデルの例

標準作業時間が二つ設定された例である。サイクルタイムの揺らぎの影響を受けずにモチーフを追跡できるように、作業モデルは、標準作業時間を中心とする正規分布の混合で定義される。

Example of operation process model generated from standard takt time

の間のセンサーデータを、標準化した後に符号化する⁵⁾。符号化では、まず、モチーフより十分短い時間のブロックごとに、センサーデータの平均値を計算することで、データを離散化する。更に、データの分布に応じて定められたしきい値に従い、ブロックごとに符号化する。例えば、各ブロックについて、平均値がしきい値 β_1 以上 β_2 以下であれば、シンボルbというように符号化し、cbaaabb...のような系列符号データを生成する。

次に、符号化されたセンサーデータについて、開始から t_{init} 間においてランダムなウィンドウ幅でランダムな長さ(幅)のモチーフを切り出し、そのモチーフの出現をパーティクルフィルターで追跡する。この追跡手法は、基本的に2.4節で説明する手法と同様であるが、ここでは符号化されたセンサーデータを用いるため、データセグメント間の距離計算には、DTWではなく符号の類似度の尺度であるハミング距離を用いる。

このような手順により、少ない計算コストで類似度計算を行うことができる。作業モデルに従って出現するこの手順により選ばれた上位 k 個のモチーフ(符号化する前の信号)から、2.4節の手法を用いて最も有用なモチーフを発見する。

2.4 パーティクルフィルターを用いたモチーフの追跡

サイクルタイムの変動に対応するために、DTWによるパーティクルフィルターを用いて詳細なモチーフの追跡を行う。そして、作業モデルに最も従うタイミングで出現するモチーフを、サイクルタイム計測に最も有用なモチーフとする。パーティクルフィルターは、サンプリング、重み付け、リサンプリングの三つのフェーズを繰り返すことで、システムの状態を推定する。開発した手法では、このパーティクルフィルターの手法に従い、逐次的にモチーフの出現を追跡する。以降では、パーティクルフィルターを用いてモチーフの出現時刻を追跡する手法について説明する。

2.4.1 初期化 パーティクルの初期化フェーズにおいて、初期の t_{init} の間のセンサーデータからランダムなウィンドウ幅でランダムな長さ(幅)に切り出されたモチーフをパーティクルとし、その時刻をパーティクルが1回目に出現した時刻として、パーティクルフィルターを用いて以降のモチーフの出現を逐次的に追跡する。

ただし、初期化が行われるのはモチーフ候補の絞り込みだけであり、絞り込まれて符号化されていない元のモチーフを、DTWによる距離計算で詳細に追跡する。

2.4.2 サンプリング あらかじめ定義した作業モデルに従い、ランダムにパーティクルをサンプリングする。 n 番目

のパーティクル x_n が i 回目に出現した時刻が $t(x_n, i)$ のとき、 $i+1$ 回目の出現時刻 $t(x_n, i+1)$ は、式(1)の状態方程式によって決定する。

$$t(x_n, i+1) = t(x_n, i) + \Delta t \quad (1)$$

Δt は作業モデルからランダムにサンプリングされた、 x_n の出現間隔の推定値である。出現する時刻 $t(x_n, i+1)$ の推定値として、注目する n 番目のパーティクルから n_s 個のパーティクルをサンプリングする。異なる標準作業時間を持つ作業工程が繰り返される場合、作業モデルは正規分布の混合で表現されるため、サンプリングされる推定値も異なる標準作業時間を基にランダムにサンプリングされる。

2.4.3 重み計算 サンプリングフェーズで、作業モデルに従ってサンプリングされた各パーティクルのスコアを、パーティクルの重みとして計算する。重み計算の概要を図3に示す。例えば、 x_n が $i+1$ 回目に出現する時刻の推定値と実際のセンサーデータを比較し、その推定値を評価する。そのため、初期化フェーズで切り出されたモチーフと、スライディングウィンドウ内にある各セグメントとの類似度の時系列データ $g_s(t)$ を利用する。ここで、 $g_s(t)$ の時刻 $t(x_n, i) - \sigma$ と $t(x_n, i+1) + \sigma$ 間に着目する。 i 回目及び $i+1$ 回目に x_n がそれぞれ時刻 $t(x_n, i)$ と $t(x_n, i+1)$ に実際に出現した場合、図3に示すように、 $t(x_n, i)$ と $t(x_n, i+1)$ を中心とするセグメントとモチーフの類似度は高くなる。理想的には、 $g_s(t)$ は、時刻 $t(x_n, i)$ と $t(x_n, i+1)$ の周辺の値が大きくなり、それ以外の値は小さくなる。そこで、この手法では、図3に示すように、時刻 $t(x_n, i)$ と $t(x_n, i+1)$ のそれぞれに中心を持つような、二

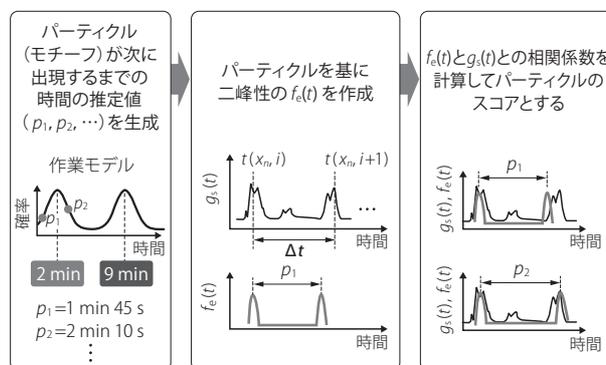


図3. 重み計算フェーズの概要

作業モデルを用いてモチーフ候補のスコアを計算して重みとする。モチーフの出現が作業モデルに従うと、重みが大きくなる。

Outline of motif candidate weight calculation phase

つの正規分布の混合関数として評価関数 $f_e(t)$ を式(2)のように定義する。

$$f_e(t) = \exp\left(-\frac{(t-t(x_n, i))^2}{2\sigma^2}\right) + \exp\left(-\frac{(t-t(x_n, i+1))^2}{2\sigma^2}\right) \quad (2)$$

この評価関数 $f_e(t)$ と、時刻 $t(x_n, i)-\sigma$ と $t(x_n, i+1)+\sigma$ 内のスライディングウィンドウごとに計算したモチーフとの類似度において、 $g_s(t)$ の相互相関を計算することで、実際にモチーフが時刻 $t(x_n, i+1)$ に出現したかを評価する。相関係数 r は、式(3)で求める。

$$r = \frac{\sum_t (f_e(t) - \bar{f}_e)(g_s(t) - \bar{g}_s)}{\sqrt{\sum_t (f_e(t) - \bar{f}_e)^2} \sqrt{\sum_t (g_s(t) - \bar{g}_s)^2}} \quad (3)$$

ただし、 \bar{f}_e 、 \bar{g}_s は、それぞれ $f_e(t)$ 、 $g_s(t)$ の相加平均である。この r が x_n の重みとなる。

2.4.4 リサンプリング 計算された重みに従いパーティクルをリサンプリングする。サンプリングフェーズでは、あるモチーフに関する一つのパーティクルから、 n_s 個のパーティクルをサンプリングしていた。ここで、これらパーティクルの出現間隔が、サイクルタイムの推定値に相当する。リサンプリングフェーズでは、その n_s 個のパーティクルの中から、一つだけを重みに応じてリサンプリングする。すなわち、リサンプリングされたパーティクルに対応する時刻が、そのモチーフが $i+1$ 番目に出現する時刻となる。

2.5 サイクルタイムの推定

2.4節で述べた処理を時刻 t_{ms} まで繰り返すことで、初期化フェーズにおいてランダムに生成したモチーフ(若しくは、符号化処理により選定されたモチーフ)の出現時刻を追跡することができる。最後に、初期化フェーズで生成したモチーフの中から、注目する作業工程のサイクルタイムを推定するために、最も適したモチーフを決定する。例えば、2.4.3項で述べた、重み付けフェーズでの重みの計算方法と同じように、評価関数 $f_b(t)$ を用意し、それを用いてモチーフの r を計算する。このとき、 $f_b(t)$ は、この方法により推定されたモチーフの出現時刻を中心とする正規分布の混合関数となる。例えば、そのモチーフが n 回出現した場合には、 n 個の正規分布の混合関数となる。そこで、時刻 t_{ms} までの $g_s(t)$ と $f_b(t)$ との r を計算し、モチーフのスコアとする。そして、最もスコアの高いモチーフを用いてこれらの処理を繰り返すことで、サイクルタイムを測定する。

3. 評価実験

3.1 データセットと評価手法

当社の製造現場で、作業者の右手首に装着した加速度センサーを用いて、組立作業時の右手の3軸加速度データを取得した。3軸の加速度データが得られるため、DTWによる類似度は、各軸で計算された距離を平均することで求める。作業者が行う作業工程は、最終組立と検査である。また、幾つかの製品を組み立てた後、それらを箱詰めする工程が存在するため、標準作業時間は140sと550sの2種類が存在する。作業者から40minのデータを取得し、その間に工程は11回繰り返された。このデータのほかにも、製造現場でデータを取得して評価した^{(3), (4)}。

この手法の精度を評価するために、各作業工程の推定されたサイクルタイムと実際のサイクルタイムとの平均絶対誤差を求めた。実際のサイクルタイムは、ビデオ撮影によって記録した。また、比較手法として、符号化によるモチーフ候補の絞り込みを行わない手法(以下、絞り込みなしの手法と略記)によっても平均絶対誤差を求めた。絞り込みなしの手法では、ランダムに15個のモチーフを選択し、DTWによるパーティクルフィルターを用いたモチーフの追跡だけを行った。これらの実験に用いた各パラメーターは、 t_{init} が140s(開発した手法だけ)、 t_{ms} が20min、 k が3、 n_s が60であった。

3.2 実験結果

開発した手法で推定したサイクルタイムの平均絶対誤差は、3.0%であった。この誤差は小さく、開発した手法は、事前の学習なしでも十分な精度を達成できた。

また、絞り込みなしの手法による推定サイクルタイムの平均絶対誤差は、3.3%であった。開発した手法は、符号化によってモチーフの候補を絞っても、モチーフをランダムに選択する絞り込みなしの手法と同等の精度を達成した。

開発した手法と絞り込みなしの手法における、各過程の計算時間を表1に示す。開発した手法は、20minのセンサーデータを処理するのに82.1sしか掛からないのに対し、絞り

表1. 開発した手法と絞り込みなしの手法による計算時間の比較

Comparison of calculation times of methods with and without selection of motif candidate

手 法	モチーフ候補の絞り込み(s)	モチーフの追跡(s)	合計(s)
開発した手法	6.0	76.1	82.1
絞り込みなしの手法	—	1,765.3	1,765.3

込みなしの手法では30 min程度掛かっている。この結果から、符号化によるモチーフの絞り込みによって計算時間を大幅に削減できた。

4. あとがき

手首に装着した加速度センサーから得られるデータに対し、教師なしで作業工程のサイクルタイムを推定する手法とその有効性について述べた。

開発した手法の精度向上と連続した長時間のサイクルタイム計測に向けて、複数の製造現場における複数の作業工程で、データを取得して評価を進めている。また、加速度センサー以外のセンサーによるデータ取得も行っており、製造現場の生産性向上につながるような、サイクルタイムの自動計測結果から更に導出することができる情報の自動抽出に向けた取り組みも行っていく。

文 献

- (1) Koskimäki, H. et al. "Activity recognition using a wrist-worn inertial measurement unit: a case study for industrial assembly lines". 2009 17th Mediterranean Conference on Control and Automation (MED 2009). Thessaloniki, Greece, 2009-06, IEEE. 2009, p.401-405.
- (2) Stiefmeier, T. et al. "Combining motion sensors and ultrasonic hands tracking for continuous activity recognition in a maintenance scenario". Proceedings of Tenth IEEE International Symposium on Wearable Computers (ISWC 2006). Montreux, Switzerland, 2006-10, IEEE. 2006, p.97-104.
- (3) Namioka, Y. et al. Automatic Measurement of Lead Time of Repetitive Assembly Work in a Factory Using a Wearable Sensor. Journal of Information Processing. 2017, 25, p.901-911.

- (4) Maekawa, T. et al. "Toward Practical Factory Activity Recognition: Unsupervised Understanding of Repetitive Assembly Work in a Factory". Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp 2016). Heidelberg, Germany, 2016-09, ACM. 2016, p.1088-1099.
- (5) Lin, J. et al. "Finding motifs in time series". Proceedings of the 2nd Workshop on Temporal Data Mining. Edmonton, AB, Canada, 2002-07, SIGKDD. 2002, p.53-68.



浪岡 保男 NAMIOKA Yasuo, Ph.D.

研究開発本部 生産技術センター グローバルモノづくり変革推進部
博士(工学) IEEE・電子情報通信学会・情報処理学会・人工
知能学会・日本データベース学会会員
Global Manufacturing Innovation Dept.



前川 卓也 MAEKAWA Takuya, Ph.D.

大阪大学大学院 情報科学研究科准教授
博士(情報科学) ACM・IEEE・情報処理学会・
日本データベース学会・電気学会会員
Graduate School of Information Science and Technology, Osaka Univ.