F-013

組み込みシステム向け異常検知方式
Anomaly Detection Method for an Embedded System

中川 慎二†
Shinji Nakagawa

1. はじめに

機械学習を用いた異常検知が，近年，注目されている[1][2][3]．従来の異常検知方式は，比較的規模の大きなシステムに適用されてきたが，今後，IoT(Internet of Things)の普及，自動運転車に代表される端末での処理高度化などを背景に，エッジでの異常検知のニーズが高まる．

本発表では，組み込み制御システムへの適用を想定した異常検知方式を提案する．正常データを用いて，クラスタリングを行い，各クラスタで構成される空間を超直方体で近似表現する．各超直方体を組み合わせて，正常空間全体を近似表現する．新たな検知対象のサンプルが得られたとき，サンプルが距離の意味で最も近い中心ベクトルに対応する超直方体内にサンプルがあるか否かで外れ値異常検知を行う．また，サンプルが距離の意味で最も近い中心ベクトルでサンプルを代表し，中心ベクトルの遷移パターンを用いて，サンプルの遷移異常を簡易的に検出する．

本方式では，正常空間を超直方体で表現するので，異常検知時の計算負荷が少なく，組み込みシステムに適する．また，クラスタ（超直方体）の数を増やすことで，正常空間の近似精度は上がるので，外れ値異常検知精度，遷移異常検知精度もクラスタ数の増加に応じて最適化することが可能である．実システムを対象に本異常検知方式を評価した結果についても述べる．

2. アプローチ

本研究の目的は，主に組み込み制御システムの動作異常を検知することである．下記を前提とした．

・過去に実績のあるデータを，正常データとする．
・過去に実績のないデータは，すべて異常データとする．

上記の前提の下，制御システムの動作異常を下記の２つに分類した．

・外れ値異常(outlier anomaly)
　正常時に期待される値から外れた状態を指す．従来からある異常検知対象の一つである．
・遷移異常(transition anomaly)
　正常時に期待される値の遷移から外れた状態を指す．

上記２つの異常を検知するアプローチを図.1 に示す．

①外れ値異常の検知
・正常動作時における制御に関するパラメータを要素とするベクトル（以下，特徴ベクトル）が，存在するベクトル空間を正常空間とする．
・異常検知対象である特徴ベクトルと正常空間からの距離の大きさに基づいて，外れ値異常判定をする．なお，本発表では，異常検知対象である特徴ベクトルが，正常空間内にあるか否かで，外れ値検知を行う．

†（株）日立製作所 :Hitachi, Ltd.

②遷移異常の検知
・正常空間を分割し，異常検知対象であるベクトルがどの分割された正常空間に属するかを決定する．
・検知対象ベクトルが属する分割正常空間もしくは分割正常空間の代表ベクトル間の遷移で，検知対象ベクトルの遷移を簡易的に代替表現し，遷移発生の異常度の大きさに基づいて，遷移異常判定をする．本発表では，異常検知対象である代表ベクトルの遷移発生頻度が０より大きいとき，正常と判定し，０のときは遷移異常と判定する．

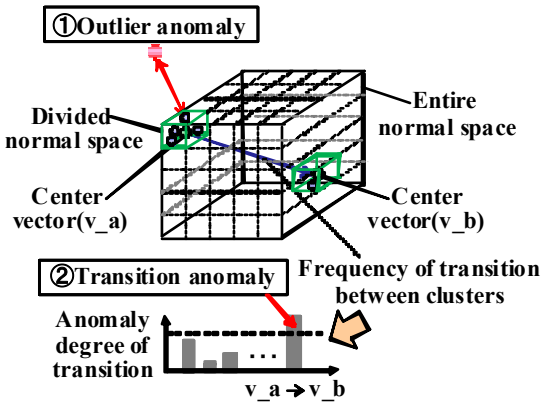

図１ アプローチ

3. 訓練フェーズ

3.1 正常空間の構築

一般に異常検知は，その他の機械学習と同様に，訓練フェーズとテストフェーズの２つのフェーズに分かれる．訓練フェーズで，正常空間を学習する．テストフェーズで学習した正常空間を用いて検知対象ベクトルの異常を検知する．

訓練フェーズの正常空間の生成方法について述べる．

i) ベースクラスタリング
　クラスタリング手法 k-means を用いて，クラスタリングを行う[4]．クラスタリング数は，目標の異常検知性能および正常検知性能が得られるように決める．k-means の中心ベクトルの初期値は，k-means++法で決める[5]．

ii) 分割正常空間の生成
　各クラスタに属する特徴ベクトルの各要素の最小値および最大値で構成される超直方体を分割正常空間とする．

iii)正常空間全体の生成
　超直方体により構成される超立体の内部を正常空間とする．

図２に，特徴ベクトルが２次元の場合の k-means によるクラスタリング結果の例を示す．図３に超直方体の組み合わせによる正常空間生成の例を示す．

Copyright © 2017 by Information Processing Society of Japan and
The Institute of Electronics, Information and Communication Engineers
All rights reserved.

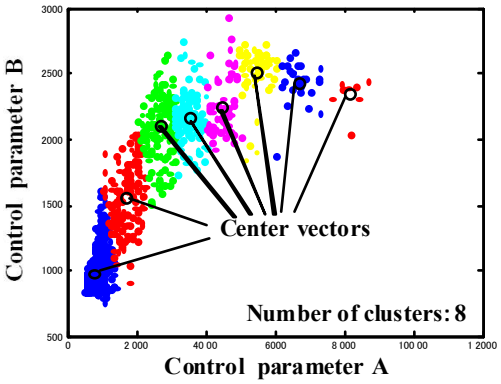

図 2 ベースクラスタリング

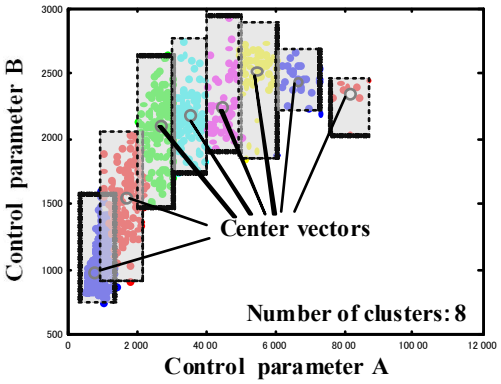

図 3 超直方体（分割空間）の組み合わせによる正常空間生成

3.2 遷移発生頻度の演算

訓練時の遷移パターン発生頻度の演算方法について述べる．

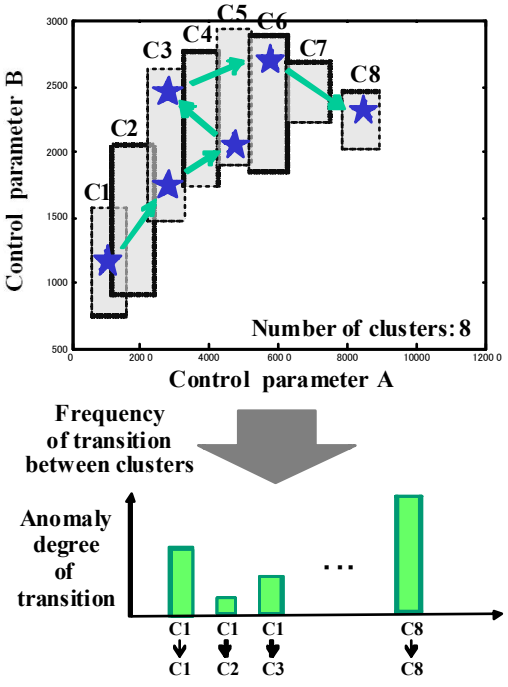

図 4 遷移発生頻度の演算

訓練時の遷移パターン発生頻度の演算方法について述べる．特徴ベクトルの時系列順の遷移の発生頻度を，特徴ベクトルが属する分割正常空間の間の遷移の発生頻度で代替演算する．図４は，特徴ベクトルが２次元の場合の特徴ベクトルの遷移の様子（上側）と分割正常空間の間の遷移発生頻度のイメージ（下側）を示している．

4. テストフェーズ

4.1 外れ値異常

テストフェーズの処理について述べる．異常検知対象の特徴ベクトルを時系列順に下記処理を実施する．

i) 検知対象の特徴ベクトルが得られたら，Ｌ２距離が，もっとも近いクラスタの中心ベクトルを選ぶ．

ii) 選んだ中心ベクトルに対応する分割正常空間の内部に，検知対象の特徴ベクトルが存在しなければ，外れ値異常と判定する．

図５は，外れ値異常の検知例を示している．青色の★は，Ｌ２距離がもっとも近い中心ベクトルに対応する分割正常空間 C7 の内部にあるので正常値と判定する．赤色の★は，Ｌ２距離がもっとも近い中心ベクトルに対応する分割正常空間 C7 の内部にないので，外れ値と判定する．

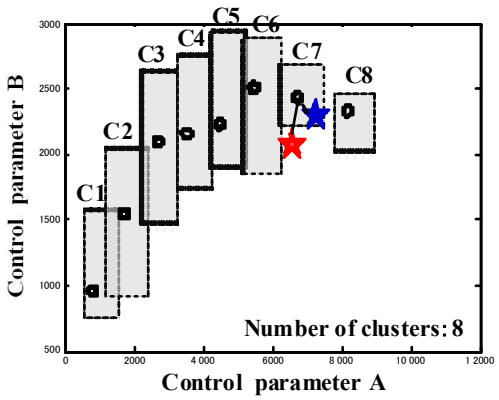

図 5 外れ値異常検知

4.2 遷移異常

i)「新しい特徴ベクトルが所属する分割正常空間（中心ベクトル）」→「前回の特徴ベクトルが所属する分割正常空間（中心ベクトル）」の遷移パターンの発生頻度が０のとき，遷移異常と判定する．

ii) 外れ値異常発生時は，遷移異常であるのは，自明であるので，外れ値異常判定を優先させる．

iii)「前回の特徴ベクトルが外れ値」で，「今回の特徴ベクトルが分割正常空間内部に存在する」ときは，遷移異常とする．今回の特徴ベクトルは，外れ値ではなく，外れ値から正常値への遷移は，訓練時には発生していないことを意味するため，遷移異常と判定する．

図６は，遷移異常の例を示している．特徴ベクトルの（番号４）から特徴ベクトル（番号５）に遷移した場合が遷移異常である．また，特徴ベクトル（番号６）から特徴ベクトル（番号７）の場合は，正常空間内での遷移ではあるが，本例においては，この遷移は訓練フェーズでは発生

Copyright © 2017 by Information Processing Society of Japan and
The Institute of Electronics, Information and Communication Engineers
All rights reserved.

していない場合（発生頻度が０の場合）を示しており，遷移異常と判定する．

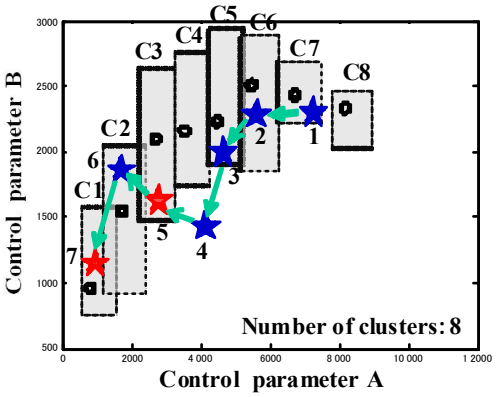

図６ 遷移異常検知

4.3 本方式の特徴

本方式による異常検知は，下記の特性を持つ．

i) テストフェーズの処理が簡便となり，組み込みシステムに向く．

・外れ値異常検知

各クラスタに属する正常ベクトルの各要素の最小値および最大値で囲まれる超直方体を分割正常空間とするので，異常検知対象の特徴ベクトルが，正常空間の内部にあるか外部にあるか外れ値異常検知処理が簡便となる．

・遷移異常検知

異常検知対象ベクトルの遷移頻度を分割正常空間（中心ベクトル）の間の遷移頻度で簡易的に代替するので，遷移ケースの組み合わせ爆発を抑制できる．

・なお，訓練フェーズは，クラスタリング処理を行うので，必ずしも組み込みシステムには向かない．サーバーなどでの集中処理が考えられる．

ii) 正常標本精度と異常標本精度の双方を向上させることが可能．

訓練フェーズで用いた正常ベクトルは，いずれかの分割正常空間内に存在するため，異常検知対象の特徴ベクトルが訓練時に用いたベクトルのいずれかと一致する場合は，必ず正常と判定され，異常と誤判定されることはない．

また，分割正常空間を組み合わせることで正常空間の全体を構成するので，分割正常空間の数を多くするほど，訓練フェーズで用いた正常ベクトルで構成される（真の）正常空間に近づき，（真の）正常空間以外の空間は，（真の）異常空間に近づく．したがって，分割正常空間の数を多くすることで，正常標本精度と異常標本精度の双方を向上させることが可能である．

iii) 目標性能，コンピューティングパワーに応じた性能設計が可能．

分割正常空間の数で検知精度を調整することが可能であり，目標性能を満足する最小の分割正常空間を選べば，計算リソースの消費を最小限に抑えることができる．

5. 実験１

5.1 実験設備と訓練用データ

小型の自律走行体の制御システムを対象とした．異常検知対象データは，自律走行体の制御操作量である目標車速と目標回転角速度とした．図７に，訓練データ（約３６０回分の走行データ）を示す．

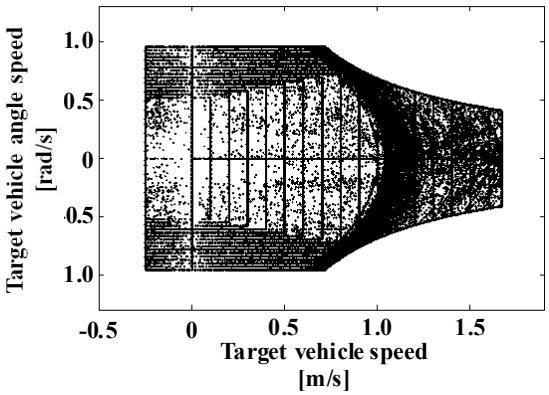

図７ 訓練データ

5.2 実験結果

図８に，分割正常空間の数が２～４００のときの正常空間を示している．分割正常空間の数が多くなるほど，訓練データの分布の近似精度が上がっていることがわかる．

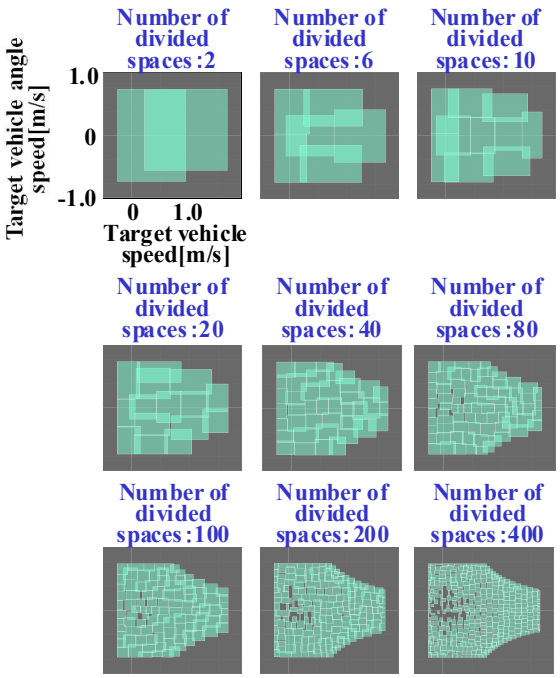

図８ 分割正常空間の数に応じた正常空間

図９は，分割空間を４００としたときの，走行データの走行回数に対する正常空間を示している．走行回数が増えるに応じて，正常空間が拡大していることがわかる．

Copyright © 2017 by Information Processing Society of Japan and
The Institute of Electronics, Information and Communication Engineers
All rights reserved.

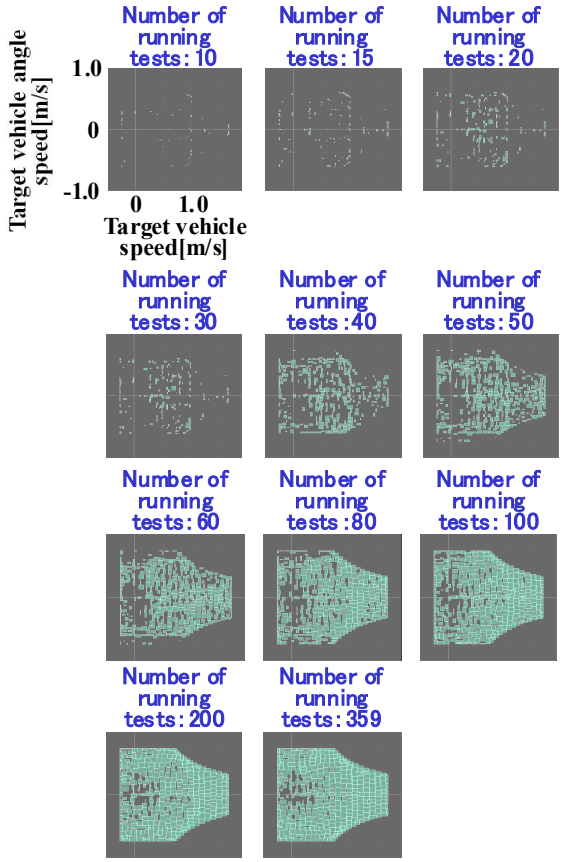

図９ 走行回数に応じた正常空間

図１０に外れ値異常検知精度の評価に用いた正常データと異常データを示している．正常データは，訓練データに用いていない正常走行時のデータを用いた．外れ値異常のデータは，図１０中に示される赤の部分であり，人工的に生成した．

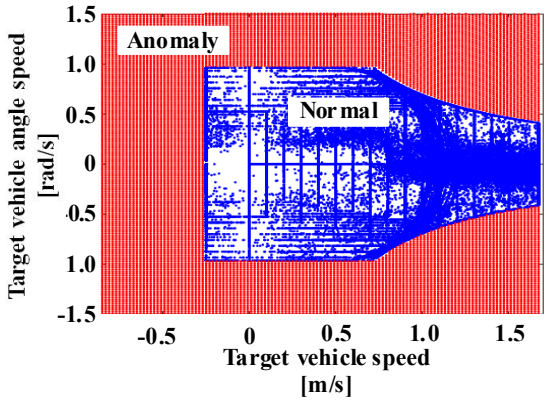

図１０ 外れ値異常検知精度評価に用いた正常データと異常データ

遷移異常検知精度に用いた正常データは，図１０に示すデータを用いた．図１０に示す正常データにおいて，正常データにはない未発生の遷移パターンを人工的に生成し，遷移異常データとした．

図１１は，分割正常空間数に対する外れ値異常検知の正常標本精度（正常データを正常と判定する率）と異常標本精度（異常データを異常と判定する率）を示している．正常標本精度は，すべての分割正常空間数で，ほぼ１００％となった．また，異常標本精度は，分割正常空間数２～１００のときは，９０～１００％で単調増加し，分割正常空間数が２００以上のときは，ほぼ１００％となった．以上，分割正常空間数が２００以上のとき，図１０に示す正常データに対する正常標本精度と外れ値異常データに対する異常標本精度の双方がほぼ１００％となった．

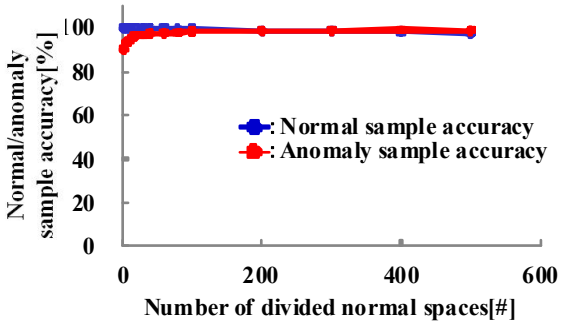

図１１ 外れ値異常検知における
正常標本精度と異常標本精度

図１２は，分割正常空間数に対する遷移異常検知の正常標本精度と異常標本精度を示している．正常標本精度は，分割正常空間数が増えるにつれ，やや減少する傾向となったが，いずれも９０％以上であった．異常標本精度は，分割正常空間数が増えるにつれ，増加する傾向となり，分割正常空間数４００と５００で，異常標本精度は，８５％以上となった．

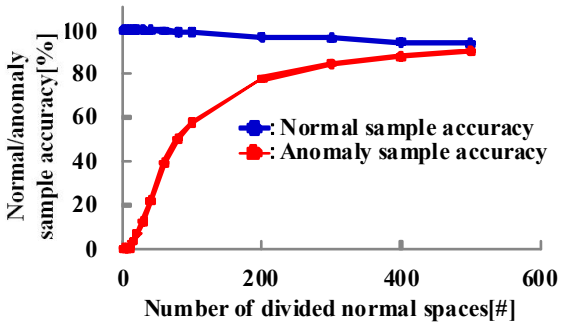

図１２ 遷移異常検知における
正常標本精度と異常標本精度

6. 実験２

6.1 実験設備と訓練用データ

図１３に，実験設備を示す．

・水の入った容器を端から端まで，容器から水をこぼさずに輸送することを目的とする試作機である．

Copyright © 2017 by Information Processing Society of Japan and
The Institute of Electronics, Information and Communication Engineers
All rights reserved.

・実際に漏水させると安全上の問題があるので，容器には蓋がしてある．容器内壁と蓋の裏側には，電極が取り付けてあり，実際に水をこぼさずに，漏水と同等の現象を検知できるようにした．
・制御系は，容器の速度をシーケンス制御する構成とした．
・異常検知するパラメータは，容器の実際の位置[mm]における目標速度[m/s]とした．すなわち，正常空間は，容器の実際の位置と目標速度で構成することにした．

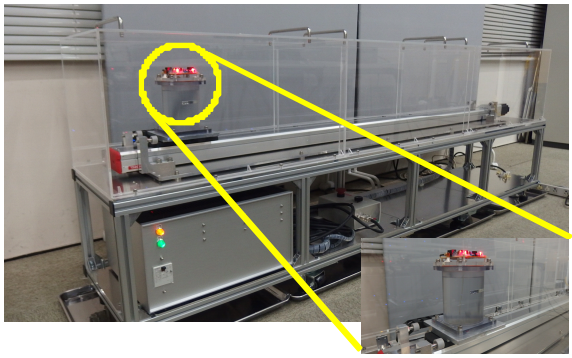

図１３ 実験設備（水容器輸送装置）

本機では，異常検知時すなわち漏水発生が予想されるときは，操作量を補正することで漏水を未然に防止する．図１４に，異常検出時のプロフィールの補正方法を示している．グラフの横軸は，容器の実際の位置，縦軸は，容器の目標速度を示している．グラフ中の青色の領域が，正常空間である．赤色のプロフィールは，制御異常時のプロフィール例である．異常を検知したとき，目標速度上下方向に法線を下ろし，正常空間と最初に交差した点をプロフィールの補正先とする．

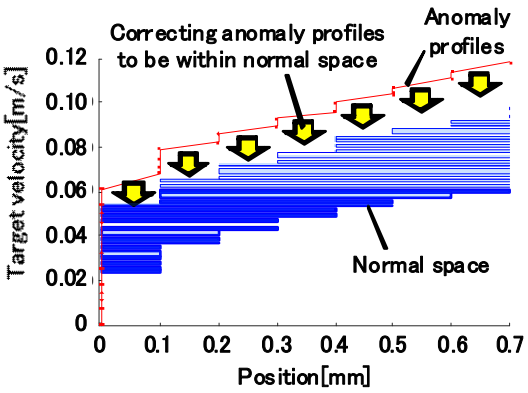

図１４ 異常検出時の補正方法

図１５は，漏水を発生させずに容器を輸送可能なプロフィールを実験により取得した結果であり，正常空間を構築するための訓練データである．グラフの横軸は，容器の実際の位置，縦軸は，容器の目標速度を示している．なお，正常空間を一定範囲にとどめるため，容器の輸送時間は，２．０２ｓに制約した．

漏水を発生させずに容器を輸送可能なプロフィールは，加速側，減速側それぞれで二つの範囲に分けられる．加速側，減速側それぞれで漏水が発生するプロフィールが現れるためである．この領域では，加速により発生する液面揺動の位相と減速開始タイミングの関係から，液面揺動が加振され，漏水が発生するためと考える．

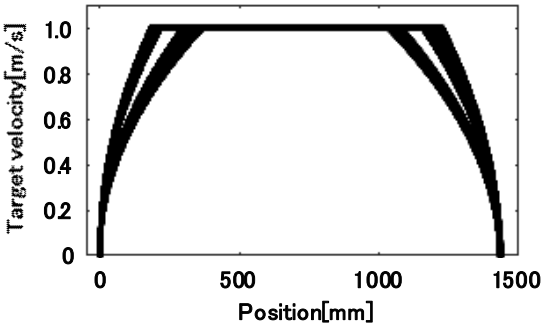

図１５ 訓練データ

6.2 実験結果

図１６に，分割正常空間の数が２～２４００のときの正常空間を示している．分割正常空間の数が多くなるほど，訓練データの分布の近似精度が上がっていることがわかる．分割正常空間が２～５０では，先述の加振現象による正常空間の分割を表現できていないが，分割空間数が１００になると，一部で正常空間の分割を表現できるようになり，以降，分割空間数が増えるにつれ，その表現精度が高くなる．分割空間数が１０００以上では，正常空間の表現精度は十分高くなっている．

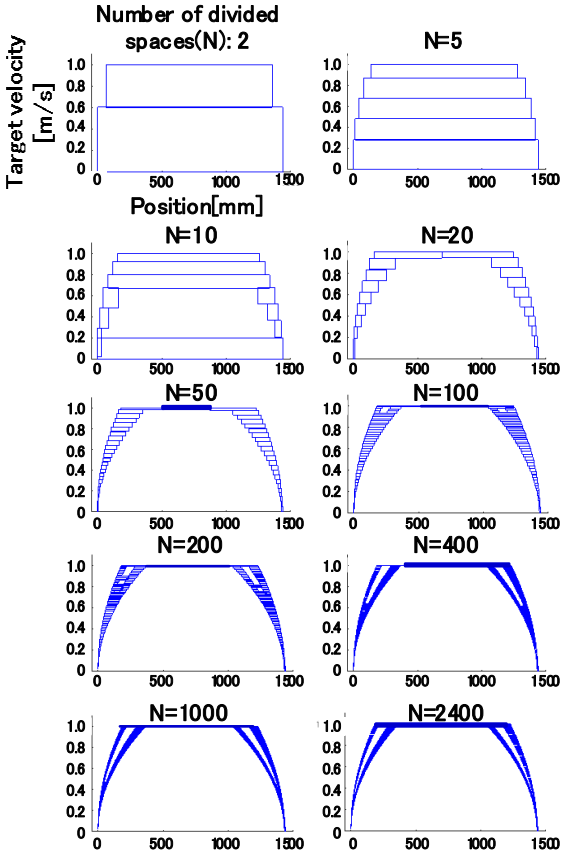

図１６ 正常空間

Copyright © 2017 by Information Processing Society of Japan and
The Institute of Electronics, Information and Communication Engineers
All rights reserved.

図１７は，正常時の異常検知アプリケーションの画面を示している．グラフの横軸は，容器の実際の位置，縦軸は，容器の目標速度を示している．グラフ中の白色の領域が正常空間である．なお，分割空間数は２４００である．青色のプロフィールは，正常時のプロフィールである．プロフィールが，正常空間内に収まっていることがわかる．

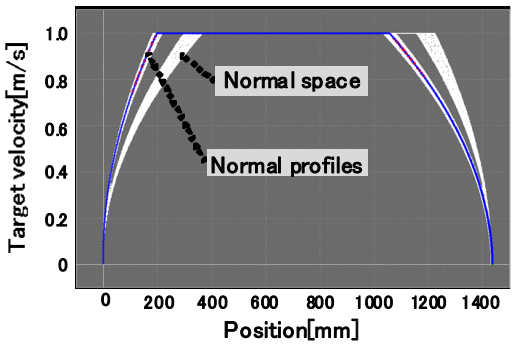

図１７ 制御プロフィール（正常時）

図１８は，加速度が大きい場合の異常検知アプリケーションの画面を示している．赤色のプロフィールは，異常プロフィールである．緑色のプロフィールは，補正後のプロフィールである．補正後のプロフィールが，正常空間内に収まっていることがわかる．

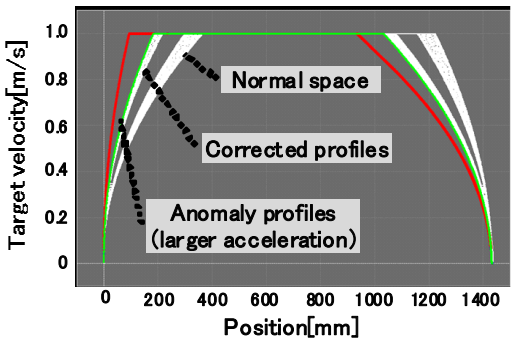

図１８ 制御プロフィール（異常時）

図１９は，制御の効果を示している．横軸は，加速時の加速度を示している．縦軸は，漏水発生率を示している．同じプロフィールで１０回，容器を移動させ，漏水が発生した移動回数が占める率を漏水発生率としている．異常検知時の補正制御なしと異常検知時の制御ありの双方の結果を示している．

＜制御なしのとき＞

図１５に示したデータに対応した結果を示している．加速度大，加速度中，加速度小の３領域で漏水が発生している．一方で，２つの正常領域では，漏水発生率０％である．

＜制御ありのとき＞

・加速度大の領域：
　８ケースすべてで，漏水発生率０％（漏水防止率１００％）
・加速度中の領域：
　６ケースの内，
　２ケースで，漏水発生率０％（漏水防止率１００％）
　１ケースで漏水発生率１０％（漏水防止率９０％）
　３ケースは，漏水発生率６０％，９０％，１００％（漏水防止率４０％，１０％，０％）
・加速度小の領域：
　９ケースすべてで，漏水発生率０％（漏水防止率１００％）

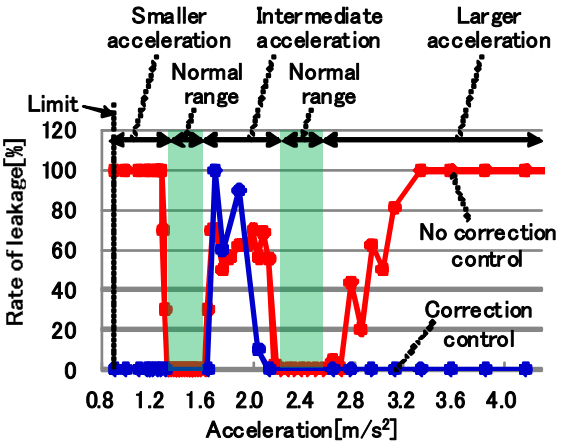

図１９ 実験結果（漏水防止率）

以上，全２３ケースの内，２０ケースで，漏水発生率１０％以下（漏水防止率９０％以上）の結果となった．一方で，漏水防止率が９０％を下回った３ケースは，いずれも加速度が中の領域である．これは，正常空間への補正方法が原因と考える．前述したように，異常検知時は，目標速度上下方向に法線を下ろし，正常空間と最初に交差した点をプロフィールの補正先とした．すなわち，目標速度の次元で，Ｌ２距離の意味でもっとも近い正常空間へ補正する方式とした．加速度が中の領域では，上下方向の双方に，正常空間が存在する．異常プロフィールによっては，補正先が上側の正常空間から下側の正常空間へ，あるいは，下側の正常空間から上側の正常空間へ変化することがあり，このとき，目標速度はステップ的に大きく変化する．これが，不適切な加速度もしくは減速度を発生させ漏水発生の原因になっていると考える．Ｌ２距離の意味でもっとも近い正常空間へ補正することが必ずしも正しくないことを示唆していると考える．

7. おわりに

（１）組み込み制御システムへの適用を想定した異常検知方式を提案した．

・過去に実績のあるデータを正常データとし，過去に実績のないデータは，すべて異常データとすることを前提とした．
・上記前提の下，制御システムの外れ値異常と遷移異常を検知する方式として，検知対象である制御パラメータを要素とする特徴ベクトルが正常時に存在する正常空間を超直方体の組み合わせで構成し，特徴ベクトルが超直方体内にあるか否かで外れ値異常を検知し，特徴ベクトルが属する超直方体間の遷移頻度で遷移異常を検知する方式とした．

Copyright © 2017 by Information Processing Society of Japan and
The Institute of Electronics, Information and Communication Engineers
All rights reserved.

・本方式では，テストフェーズの処理が簡便となり組み込みシステムに向く．訓練フェーズは，クラスタリング処理を行うので，必ずしも組み込みシステムには向かない．

（２）実システムである小型の自律走行体の制御システムを対象に本異常検知方式を評価した．

・自律走行体の目標車速と目標回転角速度を対象に性能を評価した．

・外れ値異常検知の性能評価においては，分割正常空間数が２００以上のとき，正常標本精度と異常標本精度の双方がほぼ１００％となった．

・遷移異常検知の性能評価においては，分割正常空間数が４００と５００のとき，正常標本精度と異常標本精度の双方が８５％以上となった．

（３）実システムである水の入った容器を端から端まで，容器から水をこぼさずに輸送することを目的とする試作機対象に本異常検知方式および検知結果に応じた補正制御を評価した．

・２３ケースの異常モードの内，２０ケースで漏水防止率９０％以上の結果を得た．

・加速度が大の異常モード：
８ケースすべてで，漏水防止率１００％．

・加速度が中の異常モード：
６ケースの内，２ケースで漏水防止率１００％，１ケースで漏水防止率９０％，３ケースは，漏水防止率４０％，１０％，０％．

・加速度が小の異常モード：
９ケースすべてで，漏水防止率１００％．

（４）今後の課題について述べる．

・検知パラメータの選び方
本実験では，自律走行体の目標車速と目標回転角速度を対象とした．この２つのパラメータのみで動作を決めていることが明らかだったためである．しかし，制御システムが複雑化した場合，制御システムの仕様がすべて明らかでない場合など，検知パラメータの選択も課題となる．

・正常データの与え方
本手法は，実績のあるデータを正常データと定義し，実績に応じて正常領域が広がる．正常領域で，制御システムの動作範囲が決まるので，初期状態は，一定の広がりのある正常空間を与える必要がある．市場投入前のオフライン検証データを初期正常空間として用いることが考えられる．また，市場投入後，正常空間を広げていくためには，実績データの収集が必要である．

・異常データの与え方
異常検知精度は，異常標本がないと評価できない．一般に，（正常標本数）＞（異常標本数）であり，異常を実際に起こすことは，現実的には困難なことが多い．本手法では，正常データ(実績のあるデータ)以外は，すべて異常データとし，正常空間を構築する．異常検知精度の評価においては，人工データを作成した．

・異常検知時の補正制御方法
実システムでの評価結果から，異常検知時，Ｌ２距離の意味でもっとも近い正常空間へ補正することが必ずしも正しくないことが示唆された．例えば，システム特性に応じた正常空間への補正方法の検討が必要である．

参考文献

[1] Sugiyama, M., et.al., “A density-ratio framework for statistical data processing”, IPSJ Transactions on Computer Vision and Applications, (2009)
[2] Takimoto, M., et.al., “Visual Inspection of Precision Instruments by Least-Squares Outlier Detection”, データマイニングと統計数理研究会（第１０回）, (2009)
[3] 野田，他, “高度予兆診断システムの開発”，日本機械学会動力・エネルギー技術シンポジウム講演論文集，16，39/42,（2011）
[4] MacQueen, J. B., “Some Methods for classification and Analysis of Multivariate Observations”, Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability, University of California Press, 281/297, (1967)
[5] David Arthur, “ k-means++: The advantages of careful seeding”, Proc. of the eighteenth annual ACM-SIAM symposium on Discrete algorithm, 1027/1035, (2007)

Copyright © 2017 by Information Processing Society of Japan and
The Institute of Electronics, Information and Communication Engineers
All rights reserved.