速度・変位検出形質量動的測定システム のソフトウェア†

小 野 敏 郎*・島 岡 治 夫*

Software of the Dynamic Mass-Measurement System of Displacement and Velocity Sensing Type

Toshiro ONO* and Haruo SHIMAOKA*

“Dynamic Mass-Measurement System of Displacement and Velocity Sensing-Type” (DMS-DV) consists of a spring scale, a microcomputer, and displacement and velocity sensors. In the system, to obtain the mass value of the goods very quickly in comparison with a conventional spring scale, the displacement and velocity data of the goods-plate of spring scale are data-processed in real-time by means of the software on microcomputer.

It also is an example of the software-oriented approaches based on the dynamic measurement method to meet the two contradictory requirements, “Quickness” and “Precision”.

The feasibility of the DMS-DV has already been confirmed through the theoretical analysis, the computer simulation and the experiment using the test system.

This paper describes the software which plays an important role in the utility of the DMS-DV.

Firstly, the modified version of a dynamic mass-estimation algorithm has been presented to shorten a measurement time. Secondly, the software system has been supplemented to the DMS-DV to support its operation and to meet “Usability” as well as the above two requirements.

In case of the previous algorithm, accuracy of the measured value depends on the displacement

data rather than on the velocity data. So smoothing of the velocity data by software and its positive utilization have made the present algorithm possible to shorten the measurement time within 2/3 of that which is achieved by means of the previous algorithm.

Moreover, the DMS-DV has become usable by means of the software system to support the determination, management, and renewal of the parameters and conditions for execution of the algorithm, and to assist the record and storage of the measured value.

Key Words: mass measurement, dynamic measurement method, microcomputer, spring scale, software

1. はじめに

近年、マイコンは急速な進歩を遂げ、さまざまな機器に変革をもたらした。

「速度・変位検出質量動的測定システム」(略して DMS-DV) は、マイコンと、その上で実行されるソフトウェアによって実現が可能となった新しい質量測定システムである。

本システムにおいては、被測定物を載せて振動している“ばね式はかり”の品物台の速度と変位のデータをソフトウェアによって実時間処理して、被測定物の質量を算定している。これは、測定系の動特性を積極的に利用しようとする「動的測定方式」¹⁾の考え方に基づいている。

DMS-DV の実現可能性については、理論的考察、コンピュータ・シミュレーションならびに試作システムによる測定実験によって既に検証されている^{2), 3)}。

ここでは、さらに質量動的推定アルゴリズムの改良

† 第 3 回計測自動制御学会質量・力計測部会一質量・力計測—手法・機器・システムとその応用技術シンポジウムで発表 (昭 58・9)

* 大阪府立大学工学部 堺市百舌鳥梅町 4 丁 804

* College of Engineering, University of Osaka Prefecture, Sakai

(Received July 2, 1984)

(Revised November 15, 1984)

による測定時間の短縮化，およびソフトウェア・システムの支援によるシステムの運用面での改善を試みた。

本研究の目的は，動的測定方式に基づくソフトウェアによる質量測定の「迅速さ」と「正確さ」の追求である。アルゴリズムの改良により，測定時間が0.3秒から0.19秒へと，2/3以下に短縮され，一応の限界に到達したと考えられる。

また，試作システムを用いた実験を通して，システムの「使いやすさ」も重要な条件の一つであることがわかった。すなわち，アルゴリズムで用いるパラメータの決定，更新，管理は，本システムの精度を維持するうえで重要な問題であり，これを支援するソフトウェア・システムが無ければ，システムの運用は非常に手間のかかるものになってしまう。

本論文では，アルゴリズムの改良と，ソフトウェアシステムの構成について述べ，「迅速さ」と「正確さ」に「使いやすさ」を加えた三つの条件を考慮して，DMS-DVの真の意味での実現可能性を検討する。

2. 質量動的推定アルゴリズムの改良

2.1 実用形 DV-アルゴリズム²⁾

Fig. 1 に示すように，DMS-DVのハードウェアは，基本的には，ばね式ばかり，マイコン，速度と変位の検出器，およびAD変換器から構成されている³⁾。

品物台が振動している状態において，以下に示す実用形DV-アルゴリズムによって，品物台の速度と変位のデータを実時間処理すると，被測定物の質量の推定値 $\hat{M}$ が求まる。

$$\hat{M}(2j) = F(2j)/G(2j), \quad j=1, 2, \dots \quad (1)$$

$$F(2j) \equiv mK_d \{y_v(2j) - y_v(0)\} + kK_{va} [T_{da} \{y_d(2j) - y_d(0)\} + S_d(2j)] \quad (2)$$

$$G(2j) \equiv 2K_{va}K_d g j \Delta - K_d \{y_v(2j) - y_v(0)\} \quad (3)$$

ただし， m は品物台を含めた，はかりの可動部の等価質量， k は，ばね定数， g は重力加速度， K_{va} および K_d は速度および変位検出器のゲイン， T_{da} は変位検出器の時定数である。

また， Δ はサンプリング周期であり，振動波形の観測開始時刻を t_e とすると， $\hat{M}(2j)$ は時刻 $t_e + 2j\Delta$ における被測定物質量の推定値であり， $y_v(2j)$ および $y_d(2j)$ は時刻 $t_e + 2j\Delta$ における速度および変位の観測信号である。 $S_d(2j)$ は，区間 $[t_e, t_e + 2j\Delta]$ におけるシンプソン則による変位観測信号 $y_d(2j)$ の数値積分である。

Fig. 1 Configuration of the mass-measurement system to be considered

2.2 アルゴリズムの改良

速度および変位検出器から得られる観測信号 $y_v(2j)$ および $y_d(2j)$ は観測雑音を含んでいる。

実用形DV-アルゴリズムにおいては，変位観測信号の数値積分 $S_d(2j)$ がフィルタの役割を果たすため，観測雑音の影響による質量推定値のばらつきは，観測時間 $2j\Delta$ の増加とともに減少する²⁾。

しかし，観測時間を短縮するためには，さらに観測信号の平滑化を行う必要がある。平滑化の対象となるのは，(2)，(3)式の $\{y_v(2j) - y_v(0)\}$ および(2)式の $\{y_d(2j) - y_d(0)\}$ であるが， T_{da} が十分小さい場合には，後者の影響は小さく，無視できる。

また， $y_v(2j)$ と $y_v(0)$ に対して，個別に平滑化を行うのではなく，関数 $V(i)$ を

$$V(i) = y_v(2j+i) - y_v(0+i), \quad i=0, \pm 1, \dots, \pm n \quad (4)$$

と定義する (Fig. 2 参照)。

この $V(i)$ に対して2次式の最小二乗近似平滑化⁴⁾を施し，推定値 $\hat{V}(0)$ を求め， $V(0) = y_v(2j) - y_v(0)$ のかわりに(2)，(3)式に適用する。ただし， $\hat{V}(0)$

$$\hat{V}(0) = \frac{\left(\sum_{i=-n}^n i^4 \right) \left(\sum_{i=-n}^n V(i) \right) - \left(\sum_{i=-n}^n i^2 \right) \left(\sum_{i=-n}^n i^2 V(i) \right)}{(2n+1) \left(\sum_{i=-n}^n i^4 \right) - \left(\sum_{i=-n}^n i^2 \right)^2} \quad (5)$$

である。 $n=20$ ，すなわち標本点数 $2n+1=41$ の場合

$$\hat{V}(0) = \frac{1}{22919} \left\{ 1259 \sum_{i=-20}^{20} V(i) - 5 \sum_{i=-20}^{20} i^2 V(i) \right\} \quad (6)$$

となる。

最小二乗近似平滑化をオンライン・リアルタイムで処理する際， $y_v(0)$ と $y_v(2j)$ を個別に扱うと $n > 2j - n$ の場合には，区間 $[t_e + (2j-n)\Delta, t_e + n\Delta]$ において，二つの平滑化作業を同時に実行しなければなら

Fig. 2 Time chart of a measurement

Fig. 3 Time behavior of the estimated value by the Modified DV-Algorithm in comparison with that by the Practical DV-Algorithm

ず, CPU に負担がかかる. そこで (4) 式に示す $V(i)$ に対して平滑化を施すことにした.

また, アナログ・フィルタによる平滑化も考えられるが, その場合には, 減衰, 位相等の動特性を改めて考慮しなおす必要が生ずる²⁾. したがって, ここではソフトウェアによる平滑化を選んだ.

2.3 測定実験

DMS-DV の試作システム³⁾に, 速度観測信号の最小二乗近似平滑化を付加した実用形 DV-アルゴリズム (以下では改良形 DV-アルゴリズムと呼ぶことにする) を適用し, その効果を調べた. ただし今回は, 静止時測定³⁾のみを行っている.

Fig. 3 は, 実用形 DV-アルゴリズムと改良形 DV-アルゴリズムについて, 推定誤差の平均値および標準偏差の時間的挙動を比較したものである.

横軸の観測時間は $2j\Delta$ である. これは改良形 DV-

アルゴリズムにおいては, 変位の観測時間に相当するものである. 速度観測信号の平滑化に要する部分を考慮すると, 振動波形の総観測時間は $2(j+n)\Delta$ となる (Fig. 2 参照).

なお, 被測定物の質量は, 1.6 kg, 標本数は 20 であり, 改良形 DV-アルゴリズムにおける平滑化の標本点数は 41 ($n=20$) とした. この値は, 実験の結果得られた最適な数値である.

図より, 変位観測信号の処理は同一であっても, 速度観測信号を平滑化することによって, 推定誤差の標準偏差は減少することが明らかとなった.

標準偏差の減少は, 0~0.1 秒の観測時間において顕著であり, またこの時点においては推定誤差の平均値についても若干の改善がみられる.

以上の事実から, 改良形 DV-アルゴリズムを採用することにより, 精度の向上はあまり望めないが, 測定時間を短縮できることがわかる.

なお, 改良形 DV-アルゴリズムにおいて, 推定誤差の標準偏差が, 0.03 秒付近で一度減少し, その後やや増加の傾向を示すのは, 試作システムのハードウェアに起因するものである. これが, アルゴリズムの構造や, 平滑化の手法に依存しないことは, 実験的に確かめられた.

Fig. 4 は, 観測時間を変えずに, 0.4~2.0 kg (0.2 kg ごと) の分銅を 20 回ずつ測定した際の測定値の誤差の平均値と標準偏差を示したものである. それぞれの場合の総合精度³⁾と測定条件を Table 1 にまとめておく (Fig. 2 参照).

また, 推定値 $\hat{M}$ に対して (7) 式に示す校正を施して, 測定値 $\hat{M}'$ を求めている.

$$\hat{M}' = c_0 + c_1\hat{M} + c_2\hat{M}^2 + c_3\hat{M}^3 \quad (7)$$

ただし, $c_i (i=0\sim 3)$ は, 校正係数であり, 測定条件に依存する.

図より, Case B と Case C を比較すると, 変位の観測時間は同じでも, 改良形 DV-アルゴリズムを用いると, 従来の実用形 DV-アルゴリズムに比べて, 測定誤差の平均値および標準偏差の両方が, かなり改善されたことがわかる. ただし, 測定時間は 0.02 秒長くなる.

つぎに Case A と Case C を比較してみよう.

Case A は, 実用形 DV-アルゴリズムを, 従来の測定条件³⁾で用いたものである. これに比べて改良形 DV-アルゴリズムを用いた Case C は, 測定時間が 0.3 秒から 0.19 秒に短縮されたにもかかわらず, 同等もしくはそれ以上の良い結果を示している. したがって, 精度を落とさずに, 測定時間を 2/3 以下に短縮で

Fig. 4 Measured value by the Modified DV-Algorithm in comparison with that by the Practical DV-Algorithm

きたことがわかる。

なお、Case A は、前回の静止時測定³⁾と同一の測定条件およびアルゴリズムであるが、総合精度は悪くなっている。前回のデータは'82年10月のものであり、今回のデータは'84年2月の結果である。これは、試作システムのはかり部ハードウェアの特性変化が原因であると考えられる。

2.4 考察

従来の実用形 DV-アルゴリズムでは、検出器から得られた速度観測信号をそのまま用いていた。これは推定値の精度が、最終的には変位観測信号の積分値によって保証されているからである²⁾。

しかし、観測時間をさらに短縮するためには、必然的に速度観測信号への依存の割合が高まり、これを平滑化して用いる必要が生ずる。

改良形 DV-アルゴリズムは、速度観測信号に対して最小二乗近似平滑化を行っている。この手法が最良であるとは考えないが、ソフトウェアの側面からのアプローチにおいては、Case C の結果が一応の限界であろうと思う。また、この場合、振動波形の総観測時間は、0.07 秒であり、はかりの固有周期 0.13 秒よりも明らかに短い。

3. ソフトウェア・システム

3.1 システムの運用におけるソフトウェアの利用

アルゴリズムの改良により、測定時間は 2/3 以下に短縮でき、測定の「迅速さ」と「正確さ」についての検証は、ほぼ終了した。しかしながら、測定の準備に要する時間、手間、あるいは測定データの処理、保存等の「使いやすさ」に関する検討がなされていないならば、それはアルゴリズムの実現可能性を検証したに過ぎず、真の意味で、DMS-DV の実現可能性を示したことにはならない。

測定系のハードウェアの特性を表現するパラメータの数は 6 である²⁾。その他に、振動波形の観測時間、校正係数、サンプリング周期等を合わせると、質量推定プログラムの実行に必要な数値データの総数は 20 におよぶ。このうち 6 つのパラメータおよび校正係数は、ハードウェアの特性の変動に応じて更新しなければならない。また、観測時間等も変更する場合がある。

したがって、これら数値データの決定、更新、管理は、システムの運用にとって重大な問題ではあるが、ソフトウェアによって解決可能な問題でもある。

さらに、測定データの処理および保存等もソフトウェアによって効率化できる。そこで、以上の仕事を分担する複数のソフトウェアによって、ソフトウェア・システムを構成し、DMS-DV の運用を支援すること

Table 1 Relations between total errors and the conditions of measurements

	Total error [% (2 σ , FS)]	Algorithm	Observation time for the displ. sensor [s]	Total observa- tion time [s]	Measurement time [s]
Case A	0.44	Practical DV-Algorithm	0.16	0.16	0.30
Case B	1.01	Practical DV-Algorithm	0.03	0.03	0.17
Case C	0.39	Modified DV-Algorithm	0.03	0.07	0.19

Fig. 5
Software system of
the DMS-DV

にした。

3.2 ソフトウェア・システムの構成

ソフトウェア・システムの構成を Fig. 5 に示す。

このソフトウェア・システムは、従来から DMS-DV の CPU として使用してきた汎用のパーソナル・コンピュータ (SORD M 223 MARK III) 上で実現されている。

本システムでは、オンライン・リアルタイム方式とオフライン方式の2種類の質量推定プログラムを用意している³⁾。

推定演算に必要なばね定数、観測時間等の共有データは、プログラムとは独立に、ミニフロッピー・ディスク上の FILE-C に格納されており、複数のプログラムによって利用される。

また、パラメータ同定プログラム、パラメータ・ファイル管理プログラム、パラメータ最適化プログラム、および校正係数決定プログラムの支援のもとに、FILE-C に格納された共有データは決定、管理され、さらに更新される。

FILE-C は、Fig. 6 に示すように、メインレコード、M0~M5 およびサブレコード、S0~S5 の全部で12のレコードから構成されている。

それぞれのレコードは同等であり、全部の共有データを格納しているが、更新されたデータは、日付、シリアル・ナンバーとともに M0 に書き込まれる。ただし、書き込みの前に、もとの M0~M4 の内容は、M1~M5 ヘシフトし、保持される。したがって5回前に更新されたデータまでは、メインレコード上に残っている。さらにそれ以上の保存が必要ならば、サブ

レコード、S0~S5 のいずれかか、あるいは FILE-B に転送することになる。

FILE-B は、FILE-C の内容をコピーし、保存するためのファイルである。逆に、FILE-B から FILE-C ヘデータを転送することもできる。この FILE-B もパラメータ・ファイル管理プログラムによって管理される。

データ・サンプリング・プログラムは、一定時間 (0.5 秒間) の速度と変位の観測信号をサンプルし、ミニフロッピー・ディスク上の FILE-A に格納する。

FILE-A に格納された生の速度と変位の観測データは、パラメータの決定およびオフライン質量推定に用いられる。

FILE-D は、オンライン・リアルタイム質量推定プログラムによって得られた測定値およびその測定条件を格納するためのファイルである。

3.3 ソフトウェア・システムの運用

DMS-DV の運用において、パラメータおよび校正係数の決定は重要な問題である。

測定系の数学モデル²⁾において、被測定物の質量および速度と変位の観測信号が既知であるとして、個々のパラメータを求めるための関係式を導く⁵⁾。

この関係式に基づくパラメータ同定プログラムによって、基準となる分銅を載せたときの品物台の速度と変位の観測データを処理すれば、一応パラメータの数値は求まる。ただし、これらの数値は誤差が大きく、実用に耐えない。

ここで得られた数値は FILE-C を介してパラメータ最適化プログラムに送られる。

Fig. 6 Structure of the FILE-C

このプログラムは、基準となる分銅の質量 M_i , ($i=1, \dots, n$) に対する推定値 $\hat{M}_i(2j)$, ($i=1, \dots, n$) を FILE-A の速度と変位の観測データを用いて、オフラインで求め、

$$\sum_{i=1}^n \left\{ \sum_{j=j_1}^{j_2} |\hat{M}_i(2j) - M_i| \right\}, \quad j_1 < j_2 \quad (8)$$

が最小となるように、NELDER アルゴリズム⁶⁾に従って、パラメータの最適化を行う。ただし、6つのパラメータを一度に最適化するのではなく、つぎの2組に分け²⁾、それぞれの特性を考慮して j_1 および j_2 を決めている。

- (a) 過渡状態でのみ推定値に影響をおよぼすパラメータ

$$K_{va}, T_{da}, m$$

- (b) 定常状態においても推定値に影響をおよぼすパラメータ

$$B_d, K_d, k$$

ただし、 B_d は変位検出器のバイアスである²⁾。

最適化が終了すると、その結果は FILE-C に書き込まれる。その際、校正係数には自動的に $c_1=1$, $c_i=0$, ($i=0, 2, 3$) がセットされる。

この共有データに基づき、基準となる分銅の測定をオンライン・リアルタイム方式で行い、その結果を FILE-D に書き込む。それをもとにして、校正係数決定プログラムが、その場合の測定条件に適合した校正係数を決定し³⁾、FILE-C に書き込む。

こうして、準備作業が終了し、測定開始となる。

通常は、校正係数の更新のみで、ハードウェアの特性の変動に対応できるが、必要に応じてパラメータの最適化を行わなければならない。

測定結果は、測定条件および日付、共有データ等の関連データとともに FILE-D に格納される。これにより、後になって測定結果に対して何らかの処理を施そうとした場合、非常に効率がよくなる。

なお、オフライン質量推定プログラムには、CRT に表示された既知質量の推定値の時間的挙動を参考にして³⁾、経験と勘によるパラメータのマニュアル微調整を比較的容易に行える機能が付加されている。

3.4 ソフトウェア・システムの特徴

ソフトウェア・システムを構成する各プログラムおよびファイルは、DMS-DV の実験を通して必然的に生まれたものである。当初は必要に応じて個々に作成してきたプログラム群を、全体として見渡してみれば、不完全ではあるが、一つのシステムを形成しつつあった。そこで、ソフトウェア・システムとして、より完全に機能するために、既存のプログラム間のインタフェース、すなわちデータの共有、受け渡し等を考慮して、必要なファイル、プログラムを整備することにした。

また、本ソフトウェア・システムでは、測定結果だけではなく、パラメータ、校正係数、測定条件および速度と変位の生の観測データをファイルに格納し、保存できるようになっている。

これにより、オフライン方式であれば、同一の速度と変位の観測データに対して、異なる測定条件あるいはアルゴリズムを用いることができる。

さらに、システムに異常が起きた場合、保存されたデータを用いることにより、過去にさかのぼって、その原因を系統的に調査することができる。しかも、これらのデータは、汎用のパーソナル・コンピュータのミニフロッピー・ディスク上に存在するのであるから、それを解析する為のプログラムを作成し、実行させることはたやすい。

4. 結 論

本研究は、測定系の動特性を積極的に利用することによって、質量測定の「迅速さ」と「正確さ」という二律排反する条件を同時に満足する質量測定システムの開発をめざしたものである。そしてその成果の一つが速度・変位検出形質量動的測定システム (DMS-DV) である。

今回は、DMS-DV の開発の締め括りとして、ソフトウェアの改良、整備を行い、以下に示す結果を得た。

(1) 質量動的推定アルゴリズムの改良により、測定時間を 2/3 以下に短縮することができた。

(2) システムの運用を支援するためのソフトウェア・システムの拡充により、DMS-DV は「使いやすさ」をも満足する質量測定システムとなった。

(3) (1), (2) の結果、ソフトウェアの側面からのアプローチという意味においては、DMS-DV は、ほぼ完成したと言える状況にあり、性能と運用の両面において、本システムの実現可能性を示すことができた。

本研究は、文部省科学研究費補助金、課題番号585087の補助のもとにすすめられたものである。関係各位に厚くお礼を申し上げる。

また、卒業研究として、ソフトウェアの作成に御協力をいただいた元大阪府立大学学生、松井 隆君（現在(株)日立製作所ソフトウェア工場勤務）に謝意を表す。

参 考 文 献

- 1) 小野ほか：マイクロプロセッサを利用した質量動的測定装置の試作研究，システムと制御，24-3，185/191

(1980)

- 2) 小野，島岡：速度・変位検出形質量動的推定アルゴリズム，計測自動制御学会論文集，19-4，300/307（1983）
- 3) 小野，島岡：速度・変位検出形質量動的測定システム，計測自動制御学会論文集，20-4，322/328（1984）
- 4) R. W. Hamming：ディジタル・フィルタ，宮川，今井訳，28/38，科学技術出版社（1980）
- 5) H. Shimaoka：Dynamic Mass-Measurement System of Displacement and Velocity Sensing-Type，M. Eng. Dissertation，Department of Mechanical Engineering，University of Osaka Prefecture，February（1983）
- 6) J. L. Kuester，J. H. Mize：Optimization Techniques with Fortran，298/308，McGraw-Hill Book Company（1973）