2013-3

A Steady-State Detection (SSD) Algorithm to Detect Non-Stationary Drifts in Processes

Jeff Kelly

John Hedengren
Brigham Young University, john.hedengren@byu.edu

Original Publication Citation
http://www.sciencedirect.com/science/article/pii/S0959152412002661

A Steady-State Detection (SSD) Algorithm to Detect Non-Stationary Drifts in Processes

Jeffrey D. Kelly

Industrial Algorithms, 15 St. Andrews Road, Toronto, ON, Canada, M1P 4C3

John D. Hedengren*

Department of Chemical Engineering, Brigham Young University, Provo, UT 84602

Abstract

Detecting windows or intervals of when a continuous process is operating in a state of steadiness is useful especially when steady-state models are being used to optimize the process or plant on-line or in real-time. The term steady-state implies that the process is operating around some stable point or within some stationary region where it must be assumed that the accumulation or rate-of-change of material, energy and momentum is statistically insignificant or negligible. This new approach is to assume the null-hypothesis that the process is stationary about its mean subject to independent and identically distributed random error or shocks (white-noise) with the alternative-hypothesis that it is non-stationary with a detectable and deterministic slope, trend, bias or drift. The drift profile would be typical of a time-varying inventory or holdup of material with imbalanced flows or even an unexpected leak indicating that the process signal is not steady. A probability of being steady or at least stationary over the window is computed by performing a residual Student-t test using the estimated mean of the process signal without any drift and the estimated standard-deviation of the underlying white-noise driving force. There are essentially two settings or options for the method which are the window-length and the Student-t critical value and can be easily tuned for each process signal that are included in the multivariate detection strategy.

Keywords: steady-state, stationarity, random walk with drift, white-noise, hypothesis testing, student-t.

1. Introduction

If the process or plant being monitored (passively) and/or optimized (actively) is not at steady-state then applying a steady-state model at that time is obviously not

*Corresponding author

Email addresses: `jdkelly@industrialgorthims.ca` (Jeffrey D. Kelly), `john.hedengren@byu.edu` (John D. Hedengren)

suitable given that significant accumulation or rate-of-change of material, energy and momentum violates one of the principle assumptions of the model. Applying the right model at the wrong time will result in more Type I and II errors (false positives, false negatives), biased or inaccurate parameter estimates and ultimately inappropriate decisions to be made on how to move the system to be more economical, efficient and effective. Serious violation of the steady-state assumption may result in unstable operation when on-line, real-time or closed-loop optimization is applied. Thus, correctly detecting intervals, horizons or windows when steady-state models can most likely be used is an important application. Knowing accurately in time when some processes are steady and others are unsteady can also help to identify and diagnosis potentially coincident abnormal events or symptoms noticed in other areas of the plant such as light-ends flaring, low-pressure steam venting and contaminated liquid effluents.

The subject of this work is to highlight a straightforward technique to detect periods of time in the immediate past and present when the continuous process appears to be running in a state of steady-ness or is stationary from which it is reasonable to assume that steady-state models can be implemented for the very near future. Previous work in the area of steady-state detection (SSD) is summarized by Mhamdi et. al. [1] as (a) performing a Student-t test on a linear regressed slope over the time window, (b) performing a Student-t test on two recently computed means with pooled standard-deviations from two adjacent windows and (c) performing an F-test on two recently computed standard-deviations either from two adjacent windows or from the same window but using two different filtered means. Examples of (a) can be found in Holly et. al. [2] and Bethea and Rhinehart [3], examples of (b) in Narasimhan et. al. [4] and Holly et. al. [2] and examples of (c) in Cao and Rhinehart [5] and Mansour and Ellis [6] using pre-specified exponentially-weighted filters with an interesting recursive window-based version found in Kim et. al. [7]. The method of Mhamdi et. al. [1] is somewhat similar to (a) except that it uses more sophisticated basis functions implemented in wavelet theory instead of employing a slope, trend, bias or drift component used here and as such is more difficult to apply.

Most industrial implementations of SSD use a form of (b) known as the mathematical theory of evidence (Narasimhan et. al. [4]) usually with another Student-t test on the residuals of the raw signal minus its mean divided by its standard-deviation computed over the number of data values in the window. Unfortunately, the mean and standard-deviation computed by these methods are not corrected for the drift component as is done in this SSD algorithm below. Hence, the other methods are biased (less accurate) and require more adjustment to minimize Type I and II errors. Although these techniques are easy to understand and implement, it is well-known that they require substantial and subjective tuning or calibration knowing intervals of when the plant is possibly at steady-state (Campos et. al. [8]) and is a perceived drawback. In terms of computational expense, recursive techniques can significantly reduce the computing load but since the eighties with mainframes as the supervisory computers and now with multi-core application servers this is not an issue to consider further especially for the SSD algorithm described here. The SSD algorithm uses insignificant CPU time because it only involves calculation of a mean, standard deviation, and slope.

The SSD algorithm presented in this work is also window-based and utilizes the Student-t test to determine if the difference between the process signal value minus its mean is above or below the standard-deviation times its statistical critical value. If less than, then that time instant or point is steady and if greater than, then it is unsteady where the aggregation is computed over the window approximating a probability or frequency of being at steady-state. The details of this algorithm are now presented.

2. Steady-State Detection (SDD) Algorithm

Our fundamental assumption about the behavior of the underlying system for any single process signal is to assume that it may be operating with a non-zero slope multiplied by its relative time within the window defined by the following equation for x_t as:

$$x_t = m\,t + \mu + a_t \tag{1}$$

where $m\,t$ is the deterministic drift component, μ is the mean of the hypothetical stationary process that will also equal the sample mean or arithmetic average over the time window with zero slope, and a_t is the i.i.d. random error series or white-noise sequence with zero mean and standard-deviation σ_a. Subscript t is an index that indicates the cycle at which the sample is collected while $m\,t$ refers to the slope multiplied by cycle count t. This is well-known as the "random walk with drift" non-stationary time-series found in Box and Jenkins [9] which is clearer to see when the difference x_t is first lagged or time-shifted in the immediate past x_{t-1} as:

$$x_t - x_{t-1} = m + a_t - a_{t-1} \tag{2}$$

where $a_t - a_{t-1}$ by definition has an expected value of zero with a standard-deviation of $2\ \sigma_a$. This is the simplest type of a non-stationary process and can be used to model any process with non-constant accumulation or rate-of-change of material, energy and/or momentum. An example would be a process vessel with a holdup or inventory of material where m would be non-zero with a net flow in or out of the vessel or even an unexpected leak either at the input, output or inside the vessel itself due to a loss of integrity in the system. If it is assumed that any process signal (usually a dependent variable such as a temperature, pressure or concentration) included in the steady-state detection can accumulate over time, and if this is found to be significant, then the process variable can be declared to be unsteady or non-stationary. A similar approach was taken by Kelly [10] to model the non-stationary disturbances caused by either the input or output flow, depending on if the output or input flows are the manipulated variables, of a surge vessel. A pure random walk time-series was chosen to aid in the tuning of PI controllers for improved level control known as level-flow smoothing. However, in this case the random walk with drift is used to hypothetically model potentially sustained accumulation or rate-of-change of holdup or level of the vessel over the window. This can be equally applied to any process variable which can accumulate other phenomenon such as energy or momentum with respect to time.

[1]Unbiased if the noise is i.i.d. else if there are stationary auto-regressive (AR) components (Box and Jenkins [9]) then a biased estimate will result.

By first differencing x_t it is possible to unbiasedly[1]estimate the slope m of the drift component mt as the arithmetic average of $x_t - x_{t-1}$ with n sampled values of x_t in the window which are equally spaced in time i.e., given a uniform sampling period or cycle. Slope in the linear regression can also be obtained by minimizing the sum of squared errors of the difference between the measured and model values, but this was not applied here. The slope calculation is the arithmetic mean of the first difference in x_t and is essentially a discrete-time or first-difference calculation of the first-derivative or rate-of-change of x_t with respect to time $\left(\frac{\partial x}{\partial t}\right)$ and as such m is the direct calculation or estimate of this accumulation. Obviously if there is detectable accumulation then the signal is by definition unsteady given that $\left(\frac{\partial x}{\partial t}\right)$ is found to be non-zero. The intercept μ is obtained by subtracting the term $m\,t$ from x_t when Equation 1 is rearranged to $x_t - m\,t = \mu + a_t$:

$$\mu = \frac{1}{n}\left(\sum_{t=1}^{n} x_t - m\sum_{i=1}^{n} t\right) \tag{3}$$

Now that there is an estimate of the drift slope m and the mean μ of x_t, the standard-deviation of the white-noise shocks can be estimated as:

$$\sigma_a = \sqrt{\frac{1}{n-2}\sum_{t=1}^{n}\left(x_t - m\,t - \mu\right)^2} \tag{4}$$

At this point along with a specified Student-t critical or threshold value at a particular significance level α and degrees-of-freedom n, all of the necessary information is available to test the null-hypothesis that the process signal is steady or is stationary about μ:

$$if\ \ |x_t - \mu| \leq t_{crit}\,\sigma_a\ then\ y_t = 1\ else\ y_t = 0 \tag{5}$$

The sum of y_t divided by n is a fraction related to the likelihood that the null hypothesis is false. This is the fraction of time within the window that the process or plant is deemed to be at steady-state. A fraction of 0.95 or 95% would indicate that 5% of the points are not at steady-state for example. A suitable cut-off determination of whether the process is deemed to be at steady-state depends on the application. A Student-t test could also be applied on the estimated slope m of the drift component used in previous work (see point (a) above) but this was not found to be as accurate as computing the probability over the window i.e., performing n drift-corrected residual Student-t tests and taking the arithmetic average of y_t.

Before proceeding to the results and discussion section, there are two issues that need to be addressed and they are the window-length or size and how to manage multivariate processes. The guideline for the window-length is to set it at some number of samples equivalent to greater than say three (3) to five (5) times the time-constant of the process variable divided by the sampling time-period. This implies that some prior knowledge of the process is necessary but it is more qualitative than quantitative. For example, if the time-constant is circa 30-minutes (the time to reach around 63% or $\left(1 - e^{-1}\right)$ of its steady-state gain value for a first-order process) and the sampling takes place every 2-minutes then the window-size should be between $3\text{x}\frac{30}{2} = 45$ to

$5x\frac{30}{2} = 75$ number of samples in the window. This is a typical approach used in industry for two-step[2]on-line optimization installations. The window-length should not be too short because the process will not have time to reach some level of stability and the steady-state probability will always be low precluding the use of steady-state models to help improve the profitability and performance of the plant. Too long, and multiple intervals of unsteady-state behavior within the longer window may conclude that the signal is steady when in fact it is not. These are well-known effects referred to as low/high frequency aliasing and under/over-sampling. Another possible short-coming of this approach is that false indications of steady-state may occur at the peak or valley of an oscillating process. With the horizon window centered over the peak, a slope of $m = 0$ and points within the confidence interval of Equation 5 will result. In these cases, the SSD will indicate periodic acceptance proportional to the frequency of the oscillation.

To manage multiple process signals where collectively they determine whether a system is steady, the same approach is used as found in the gross-error detection literature (Narasimhan and Jordache [11]) which was also equivalently presented in Mansour and Ellis [6] to handle multivariate systems. Essentially, the individual significance level α' is reduced or corrected from the overall significance level $alpha$ derived from the well-known Sidak inequality as:

$$\alpha' = 1 - (1-\alpha)^{\frac{1}{k}} \tag{6}$$

where k is the number of key process variables selected to be included in determining if the process or plant is steady or exhibits some level of stationarity where a number between three (3) and thirty (30) is reasonable. This means that α' will be smaller compared to α if k is greater than 1.0 and will result in a larger Student-t critical value for the same number of degrees-of-freedom n for an individual signal. The types of process signals included in the set of key variables should be some mix of manipulated (independent) and controlled (dependent) variables. Although strictly speaking, the k variables should be independent from each other for Equation 6 to be valid, it is a fair correction for these purposes to at least address multivariate systems in some way. Using several key process signals is also useful to identify individual signals which if always steady when the rest are unsteady, can be an indication that its window-length is too long. Or, if it is always unsteady when the rest are steady, may imply that its window-length is too short.

For this study, the algorithm is applied to the data in offline batch segments as demonstrated in the example applications. In these examples, there is no overlap of the analyzed time horizons. When the null hypothesis is not rejected, all of the points in the time window are deemed to be at steady-state. An alternative application for real-time systems is to process the data online as new measurements arrive. This moving horizon approach would enable real-time monitoring of the process steadiness. Additionally, multiple horizons could be processed at each sampling interval to determine the degree of steadiness over multiple time periods.

[2]Two-step meaning that there is first an estimation run which performs parameterization and reconciliation before the optimization run similar to "bias-updating" in on-line model-based control applications.

3. Results and Discussion

The testing of this SSD algorithm entails simulating a process signal with a mean of zero $(\mu = 0)$ and superimposed white-noise (a_t) generated from the code found in Ahrens et. al. [12] with a standard-deviation specified as 1.0 $(\sigma_a = 1)$ for simplicity. A window-size of 120 samples or time-periods is used where the sampling instant is assumed to be one-minute in duration simulating two-hours of real-time. Three disturbance model structures were used to generate the data sequences or time-series which are a stationary periodic cycle with $m\,sin(t) \neq 0$, a non-stationary drift with $m\,t \neq 0$ and a stationary stochastic process with $ARMA\,(p = 1, q = 1)$. For the first two disturbance models involving m the value value is varied from 0.001, 0.01, 0.1, 1.0 and 10.0 in order to assess the sensitivity to the signal-to-noise ratio which also varies from 0.001 to 10.0 given the fixed choice of $\sigma_a = 1$. For the auto-regressive part of the model found below in Equation 7, the ϕ_1 parameter is varied within the set of 0, -0.7, 0.9, -0.95 and -0.99 and for the moving-average part the θ_1 parameter is fixed at either 0 or -0.5.

$$x_t = m\,t + \mu + \frac{1+\theta_1 z^{-1}}{1+\phi_1 z^{-1}} a_t \tag{7}$$

where z^{-1} represents the lagging of one sampling instant in the past. As ϕ_1 approaches 1.0 then this becomes the most basic form of a non-stationary process (random walk) and when both ϕ_1 and θ_1 are zero then it reduces again to Equation 1. In order to confirm the standard-deviation estimate of the white-noise $(\sigma_a = 1)$ with $m = 0$, and $\theta_1 = 0$, the window-lengths are varied to 120, 1200, 12000 and 24000 yielding 1.208, 1.034, 1.010 and 0.999 respectively. Since these estimates are close to 1.0, this confirms that the driving force for the simulation is sufficiently distributed as random error. It also verifies that the calculation of white-noise standard-deviation found in Equation 4 is acceptable as well. Tables 1 and 2 show the simulated probabilities in parentheses at two different Student-t critical values of 2.0 and 3.0. The value 2.0 typically represents a 5% significance level and 3.0 is typical of a 0.5% significance. Table 1 using $m\,sin(t)$ is purposely chosen to be a stationary but cyclic deterministic type of process to show that the SSD algorithm has no reason to reject the null-hypothesis that the system is at steady-state although the signal is oscillating within the window but it is not static. All of the cells of Table 1 are sufficiently close to 95% and 99% confirming that the process signal is statistically stationary.

Using the same random seed as for Table 1, Table 2 exhibits the same results for the first row as in Table 1. The non-stationary disturbance is detected to be unsteady when m is greater than 0.01 with only white-noise (second column) and as colored-noise $(ARMA(1,1))$ is added unsteady-state operation is detected for m as low as 0.001 when the colored-noise also tends to approach non-stationarity (fifth and sixth columns).

The sensitivity of the SSD to identify unsteady-state activity when a drift is injected into the signal response has been demonstrated at least for the white and colored-noise series considered here. As the drift component magnitude increases it gets easier for the technique to declare the system unsteady (low probability of being steady) especially when $m\,t$ is consistently near or above the standard-deviation of the white-noise driving force. This is easily seen in Figure 1 where both $m = 1$ (solid line) and $m = 0.1$ (dotted line) are plotted with only white-noise. The larger m exhibits an obvious drift

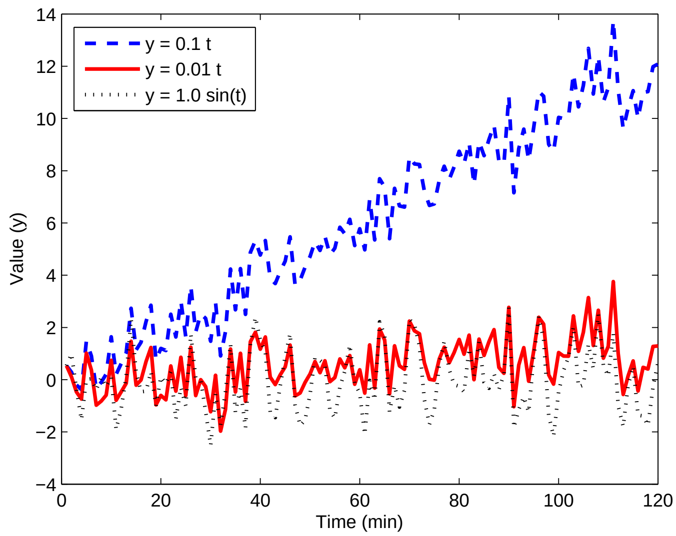

Figure 1: Plot of $1.0 sin(t)$, $0.01\,t$, and $0.1\,t$ over the window with $n = 120$.

Table 1: Simulated probability (%) results with $m\,sin(t)$ using two Student-t critical values.

m	$\theta_1 = 0.0$ $\phi_1 = 0.0$	$\theta_1 = -0.5$ $\phi_1 = -0.7$	$\theta_1 = -0.5$ $\phi_1 = -0.9$	$\theta_1 = -0.5$ $\phi_1 = -0.95$	$\theta_1 = -0.5$ $\phi_1 = -0.99$
0.0	(95.8,100)	(95.8,100)	(95.8,100)	(95.0,100)	(97.5,100)
0.001	(95.8,100)	(95.8,100)	(95.8,100)	(95.0,100)	(97.5,100)
0.01	(95.8,100)	(95.8,100)	(95.8,100)	(95.0,100)	(97.5,100)
0.1	(96.7,100)	(96.7,100)	(95.8,100)	(95.0,100)	(96.7,100)
1.0	(97.5,100)	(97.5,100)	(95.8,100)	(97.5,100)	(96.7,100)
10.0	(100,100)	(100,100)	(100,100)	(99.2,100)	(99.2,100)

Table 2: Simulated probability (%) results with mt using two Student-t critical values.

m	$\theta_1 = 0.0$ $\phi_1 = 0.0$	$\theta_1 = -0.5$ $\phi_1 = -0.7$	$\theta_1 = -0.5$ $\phi_1 = -0.9$	$\theta_1 = -0.5$ $\phi_1 = -0.95$	$\theta_1 = -0.5$ $\phi_1 = -0.99$
0.0	(95.8,100)	(95.8,100)	(95.8,100)	(95.0,100)	(97.5,100)
0.001	(95.8,100)	(95.8,100.0)	(95.0,99.2)	(87.5,97.5)	(77.5,89.2)
0.01	(92.5,98.3)	(58.3,81.7)	(38.5,52.5)	(31.7,40.8)	(23.3,36.7)
0.1	(20.0,26.7)	(0.1,10.8)	(0.0,10.0)	(0.0,10.8)	(0.0,19.2)
1.0	(0.0,0.0)	(0.0,0.1)	(0.0,0.1)	(0.0,0.1)	(0.0,17.5)
10.0	(0.1,0.1)	(0.0,0.0)	(0.0,0.0)	(0.0,0.1)	(0.0,16.3)

up whereas the smaller m has a more subtle trend up and cannot be declared as unsteady although it does have a lower probability than $m = 0.001$ in column two indicating an increased residual violation of the stationarity assumption. The term $1.0 sin(t)$ (second dotted line) is also plotted to not only show it is stationary as confirmed in Table 1 but also to highlight the slight but noticeable drift up of the solid line as the sample number increases. And expectantly, when the window-length is increased from 120 to 480 samples probabilities of (49.0,70.0) are obtained confirming that the signal is non-stationary though requiring more time to statistically detect that it is unsteady. This is not an unusual observation given that it is well-accepted that subtle perturbations require more sample or data points.

3.1. Multivariate Case Study

A simple model of a jacketed continuously stirred tank reactor (CSTR) is used to demonstrate the SSD algorithm. The problem has been used extensively in the literature to benchmark new techniques because of some unique characteristics that pose a variety of desirable challenges [13]. One challenge is the nonlinearity of the system due to the exothermic first-order reaction. The exponential dependency on temperature causes order of magnitude differences in reaction rates depending on the reactor temperature. Above a jacket temperature of approximately 305 K, the CSTR enters a sustained oscillation of temperature run-away followed by reaction quenching and concentration build-up of A. Once the concentration C_A reaches a sufficiently high level the temperature runs away, leading to the next cycle. For this case study, the CSTR is perturbed by adjusting the jacket temperature T_c but does not become unstable as mentioned above. The nonlinear model demonstrates that the SSD algorithm is applicable to multivariate processes with strong nonlinearities.

The CSTR model consists of a feed stream of pure A at concentration $C_{A,i}$ and inlet temperature T_i. The reactor is well mixed and produces product B with an exothermic first-order reaction. The reactor temperature and extent of reaction are controlled by manipulating the cooling jacket temperature T_c with negligible dynamics for the speed of cooling jacket temperature response. The variables for this CSTR model are shown in Table 3 and the equations are shown in Table 4.

Table 3: CSTR Parameters and Variables

Manipulated variable			
Symbol	Description	Nominal Value	Units
T_c	Jacket Temperature	300	K
State variables			
Symbol	Description	Nominal Value	Units
C_A	Concentration of A in the reactor	0.877	$\frac{mol}{m^3}$
T	Temperature of the reactor	324.48	K
Other parameters			
Symbol	Description	Value	Units
$C_{A,i}$	Concentration of A in the feed	1.0	$\frac{mol}{m^3}$
C_p	Heat capacity of the liquid	0.239	$\frac{J}{kg\ K}$
E_a	Activation energy	$7.28e4$	$\frac{J}{mol}$
ΔH_r	Energy of reaction	$5x10^4$	$\frac{J}{mol}$
k_0	Pre-exponential factor	$7.2e10$	$\frac{mol}{m^3\ min}$
R	Universal gas constant	8.31451	$\frac{J}{mol\ K}$
ρ	Mixture density	1000.0	$\frac{kg}{m^3}$
T_i	Feed temperature	350.0	K
q	Feed flow rate	100.0	$\frac{mol}{min}$
V	Volume of the reactor	100.0	m^3

Table 4: CSTR Model Equations

Component balance on A

$$V \frac{\partial C_A}{\partial t} = qC_{A,i} - qC_A - k_0 C_A V \exp\left(-\frac{E}{RT}\right)$$

Energy balance

$$\rho C_p V \frac{\partial T}{\partial t} = \rho C_p q \left(T_i - T\right) + \Delta H_r k_0 C_A V \exp\left(-\frac{E}{RT}\right) + UA\left(T_c - T\right)$$

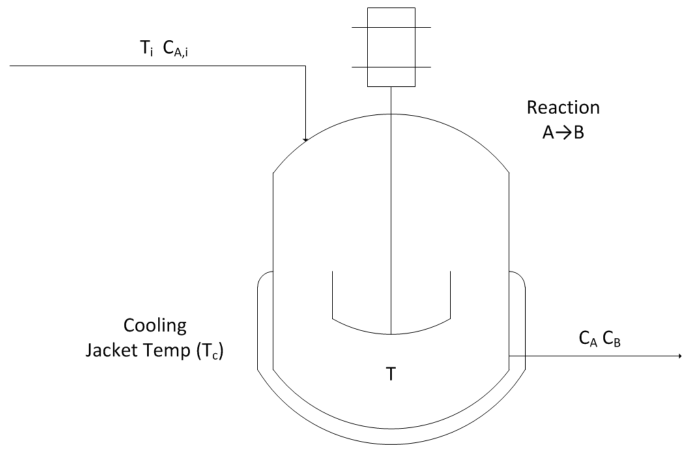

Figure 2: Diagram of the exothermic CSTR with first-order reaction kinetics.

The process simulation records data every 1-second over a total time of 40-minutes for a total of 2400 samples. A step test determined that the time constant is approximately 1-minute for C_A and 45-seconds for T. According to the guidance provided earlier, a window of 3 to 5 time constants (using the dominant time constant) is selected for analyzing the probability that the process is at steady-state. In this case, a time window of 5 time constants or 5-minutes is selected. Each time window includes 300 samples for both C_A and T. In the prior examples, the Student-t critical values of 2.0 and 3.0 were used to determine the steady-state probabilities. Because this system involves more than one variable, the Sidak inequality suggests Student-t values of 2.25 and 3.04 for the 5% and 0.5% significance levels, respectively. The cooling temperature is initially lowered from 300K to 290K for a period of 10-minutes followed by a step back to 300K for another period of 10-minutes. Following these step changes, T_c begins oscillating with a period of 3-seconds for 7-minutes before returning to the constant value of 300K for the remainder of the total 40-minutes. Random state $(\sigma(\omega_{C_A}) = 0.005, \sigma(\omega_T) = 0.05)$ and measurement noise $(\sigma(\nu_{C_A}) = 0.02, \sigma(\nu_T) = 0.2)$ are added at each sample point after the equations in Table 4 are integrated forward in time as shown in Equation 8.

$$x[t+1] = f(x[t], u[t]) + \omega \tag{8a}$$

$$y[t] = g(x[t], u[t]) + \nu \tag{8b}$$

with $x[t]$ and $y[t]$ being the state and measurement vectors, respectively, for both C_A and T. The vector $u[t]$ includes all exogenous inputs and t is the cycle index. Functions f and g are the nonlinear state and measurement functions with g simplifying to $(C_A\ T)^T$ for this example problem.

Windows 2, 7, and 8 have the highest probability $(> 90\%)$ of being at steady-state for both C_A and T above the minimum probability limit. Windows 4, 5, and 6 have either C_A or T greater than a 90% probability to a 5% significance level (first number

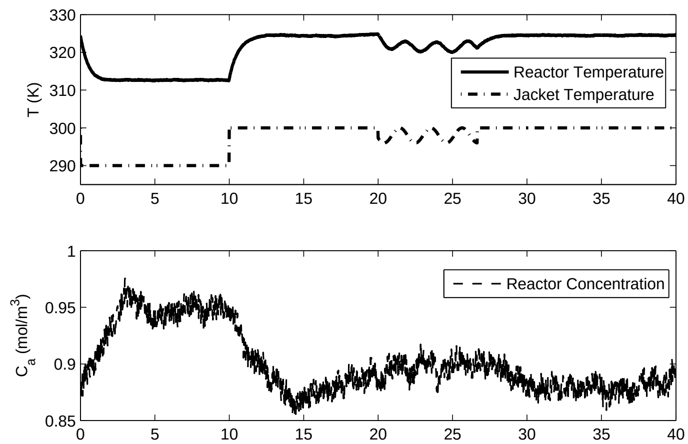

Figure 3: Simulated operational data for the CSTR. Periods of unsteady behavior in reactor concentration and temperature are observed due to steps and sinusoidal fluctuations in the jacket cooling temperature.

Table 5: Simulated probability (%) results with two Student-t critical values. The first probability represents a 5% significance level while the second represents a 0.5% significance.

Index	Time Period	Probability of C_A at Steady-State	Probability of T at Steady-State
1	0 to 5-minutes	**(56.7,80.0)**	**(17.7,100.0)**
2	5 to 10-minutes	(99.3,100.0)	(100.0,100,0)
3	10 to 15-minutes	**(28.3,39.7)**	**(37.3,100.0)**
4	15 to 20-minutes	(86.7,96.3)	(100,0,100.0)
5	20 to 25-minutes	(97.7,100.0)	(61.3,100.0)
6	25 to 30-minutes	(54.3,78.3)	(100.0,100,0)
7	30 to 35-minutes	(99.3,100.0)	(100.0,100.0)
8	35 to 40-minutes	(97.7,100.0)	(100.0,100.0)

in the parenthesis). The other two time windows (1 and 3 bolded) are deemed to not be at steady-state because both C_A and T fail to meet a minimum probability of steadiness $(\leq 90\%)$. These results are also visually consistent with Figure 3 as the two largest step changes occur in time windows 1 and 3.

3.2. Scale-up to Large-Scale and Complex Systems

One concern with any data analysis technique is the scale-up to large-scale systems. In this regard, the CPU time requirements for this SSD algorithm are negligible because it only involves calculation of a mean, slope, and standard deviation for each measurement. For practical purposes, it may be reasonable to select an appropriate cut-off probability value such as 90% for instance and/or to take an average of the two low and high probability estimates (corresponding to the low and high Student-t critical values) and apply the cut-off to this average. This is left as an implementation issue where it can be used to assist in the tuning or aligning of the quantitative results with the qualitative expectations. Although the details cannot be released, an application using this SSD technique been implemented in a fully integrated oil-refinery where SSD was considered to be a key plant, process or performance indicator (KPI) and was used to help isolate temporal root causes to process incidents.

An additional application of this technique is in identification of historical data windows that are at steady-state. This identification is useful to select data sets for parameter identification with steady-state mathematical models. When processing large amounts of data, this identification typically yields many data windows that are deemed acceptable for parameter estimation. Taking similar data sets for the parameter fit generally leads to poor results because there is not enough data diversity to fit parameters in nonlinear relationships. One example of this is that lack of temperature variability in a reactor data does not allow activation energies to be identified because of the co-linear relationship with the pre-exponential factor as shown in Table 4. If the temperature data varies, a tighter confidence interval can be obtained for both E_a and k_0. The steady-state identification procedure shown in this work can be applied to an optimization problem with an objective to obtain the best limited number of diverse data sets from a potential candidate pool.

Even though this technique is applied in time blocks, it can also be applied in a time shifted approach to signal plant steadiness or unsteadiness on a continual and real-time basis. For example, if a new sample is obtained every second, the past 3 to 5 time constants could be used to determine the probability that the process is currently at steady-state.

4. Conclusion

Presented in this work is a straightforward technique to effectively detect intervals or windows of steady-state operation within continuous processes subject to noise. This detection is critical in applications that rely on steady-state models for data reconciliation, drift detection, and fault detection. The algorithm has minimal computing requirements involving statistical estimates and has only two settings to specify i.e., the window-length and the Student-t critical value. Multivariate systems can be easily handled by including several key process signals and adjusting the critical Student-t statistic accordingly. Finally, the benefit of detecting windows of steady-state behavior in a plant with multiple interacting major processing units for example can be useful even by itself without executing on-line steady-state models for monitoring or optimization.

References

[1] A. Mhamdi, W. Geffers, F. Flehmig, W. Marquardt, Desalination and Water Resources: Thermal Desalination Processes, Vol. 1, EOLSS, 2010, Ch. On-line optimization of MSF desalination plants, pp. 136–162.

[2] W. Holly, R. Cook, C. Crowe, Reconciliation of mass flow rate measurements in a chemical extraction plant, Can. J. Chem. Eng. 67 (1989) 595–601.

[3] R. Bethea, R. Rhinehart, Applied Engineering Statistics, Marcel Dekker Inc., New York, 1991.

[4] S. Narasimhan, C. Kao, R. Mah, Detecting changes of steady states using mathematical theory of evidence, AIChE J. 33 (1987) 1930–1932.

[5] S. Cao, R. Rhinehart, An efficient method for on-line identification of steady-state, J. Proc. Cont. 5 (1995) 363–374.

[6] M. Mansour, J. Ellis, Methodology of on-line optimization applied to a chemical reactor, Appl. Math. Modell. 32 (2008) 170–184.

[7] M. Kim, S. H. Yoon, P. Domanski, W. Payne, Design of a steady-state detector for fault detection and diagnosis of a residential air conditioner, Int. J. Refrig. 31 (2008) 790–799.

[8] M. Campos, H. Teixeira, F. Liporace, M. Gomes, Challenges and problems with advanced control and optimization technologies, in: 7th IFAC Intern. Symp. Adv. Contr. Chem. Proc., Istanbul, Turkey, 2009.

[9] G. Box, G. Jenkins, Time-Series Analysis: Forecasting and Control, Revised Edition, Holden Day, Oakland, California, 1976.

[10] J. Kelly, Tuning digital pi controllers for minimal variance in manipulated input moves applied to imbalanced systems with delay, Can. J. Chem. Eng. 76 (1998) 967–974.

[11] S. Narasimhan, C. Jordache, Data Reconciliation and Gross Error Detection: An Intelligent Use of Process Data, Gulf Publishing Company, Houston, TX, 2000.

[12] J. Ahrens, K. Kohrt, U. Dieter, Algorithm no.599, ACM-Trans. Math. Soft. 9 (1983) 255–257.

[13] J. Hedengren, T. Edgar, Approximate nonlinear model predictive control with in situ adaptive tabulation, Computers and Chemical Engineering 32 (2008) 706–714.