Identification of Zone and Building Transient Thermal Response Models

Abstract. A test zone that is subjected to variable ambient temperature and variable heat rate was monitored for several weeks. The boundary conditions and the zone temperature response were recorded at 5-minute intervals during September and early October 2003. In this report, first and second-order transient thermal response models are identified. Two special approaches are used to obtain physically realistic and invertible models more reliably than by the conventional formulation and application of ordinary least squares. The test room's thermal properties are documented in Appendix A. Appendix B describes a canonical form for the general discrete-time transfer function model of a conduction (linear diffusion) dominated thermal system. Appendix C contains the model identification code. A representative fragment of the data file input to the identification codes appears in Appendix D.

Model Synthesis. Heat transfer and storage in a wall or building structure is governed by the diffusion (Fourier's) equation:

$$\rho c_p \frac{\partial T}{\partial t} = k\nabla^2 T \quad (3\text{–}D)$$
$$\rho c_p \frac{\partial T}{\partial t} = k\frac{\partial^2 T}{\partial x^2} \quad (1\text{–}D) \tag{1}$$

where k=conductivity (assumed isotropic) and ρc_p=thermal capacitance are the local material properties of interest. In steady-state, (1) reduces to Laplace's equation:

$$k\nabla^2 T = 0 \quad (3\text{–}D)$$
$$k\frac{\partial^2 T}{\partial x^2} = 0 \quad (1\text{–}D) \tag{2a}$$

to which boundary conditions are applied after integration to obtain heat flux (W/m^2):

$$\vec{q} = -k\nabla T \quad (3\text{–}D)$$
$$q_x = -k\frac{dT}{dx} \quad (1\text{–}D) \tag{2b}$$

Applied to a wall (or an enveloping surface) in steady state:

$$Q_z = u(T_w - T_z) \tag{3}$$

where T_w is a uniform exogenous temperature[1], T_z, is zone[2] temperature (Seem 1987), and Q_z is net heat input (flux integrated over area, Watts) to the zone and $u = \Sigma UA$ (W/K) is the parallel conductance of all subareas defined on the envelope.

With multiple walls:

$$Q_z = \sum_{w=1}^{W} u_w (T_w - T_z) \tag{4}$$

Note that sum of T-coefficients equals zero explicitly in this, the steady state, case.

Now consider the discrete-time, linear, time-invariant dynamic system with one wall, w:

<<insert state-space finite-difference eqns for homogenous slab with 1-D flux here>>

$$B^{0:n}(\phi_z, Q_z) + B^{0:n}(\theta_w, T_w) - B^{0:n}(\theta_z, T_z) = 0 \tag{5}$$

where the coefficient vectors each have $n+1$ elements:

---

[1] Several exterior surface or sol-air temperatures may be included as forcing functions

[2] Seem's linearized star model approximates the weighted mean of air and mean-radiant temperatures

$\phi_z = [\phi_{z,0}\ \phi_{z,1} \cdots \phi_{z,n}]$
$\theta_z = [\theta_{z,0}\ \theta_{z,1} \cdots \theta_{z,n}]$
$\theta_w = [\theta_{w,0}\ \theta_{w,1} \cdots \theta_{w,n}]$,
and the time-varying condition and response vectors have corresponding elements:
$Q_z = [Q_z(t)\ Q_z(t\text{-}1) \cdots Q_z(t\text{-}n)]$
$T_z = [T_z(t)\ T_z(t\text{-}1) \cdots T_z(t\text{-}n)]$
$T_w = [T_w(t)\ T_w(t\text{-}1) \cdots T_w(t\text{-}n)]$,
Each backshift polynomial term in (5) expands to a sum, as follows,

$$B^{0:n}(x, y) = \sum_{k=0}^{n} x_k y(t-k),$$

where t is now, and hereafter, defined as the discrete time index.
It is convenient, when disassociating the 0-lag terms of (5), to use a more general form of the backshift polynomial notation in which the index of summation starts not at 0 but at i:

$$B^{i:n}(x, y) = \sum_{k=i}^{n} x_k y(t-k).$$

Note that one coefficient in can be assigned arbitrarily; it is handy[3] to let ϕ_z = -1 so that (5) becomes:

$$Q_z = B^{1:n}(\phi_z, Q_z) + B^{0:n}(\theta_w, T_w) - B^{0:n}(\theta_z, T_z) \qquad (6)$$

This form of the sought-after response function predicts heating or sensible cooling load, Q_z, given zone temperature, T_z, and one or more exogenous temperatures or sol-air temperatures, T_w.

To satisfy the steady state response model (4) we must have:

$$\frac{\sum_{w=1}^{W}\sum_{k=0}^{n}\theta_{w,k}}{\sum_{k=0}^{n}\phi_{z,k}} = \frac{\sum_{k=0}^{n}\theta_{z,k}}{\sum_{k=0}^{n}\phi_{z,k}} = \sum_{w=1}^{W} u_w \qquad (7)$$

In the steady-state formulations (3, 4) the sum of temperature coefficients equals zero by definition but in the discrete time (DT) dynamic model formulation (5) the constraint $\Sigma_w\Sigma_k\theta_{w,k} = \Sigma_k\theta_{z,k}$ must be consciously enforced.

The DT model (5) can be evaluated recursively for Q_z as in (6), or for T_z as follows:

$$\theta_{z,0}T_z = B_0^n(\phi_z, Q_z) + B_0^n(\theta_w, T_w) - B_1^n(\theta_z, T_z) \qquad (8)$$

(Equivalently, T_z = RHS expression divided by $\theta_{z,0}$)

An expression for Q_z like (6) is known as a conduction transfer function, or CTF (Walton 1983). We will call the corresponding expression for T_z (8) the *inverted CTF* or iCTF.

There are additional conditions (constraints) the coefficients must satisfy for these models to be physically (thermodynamically) plausible. To see this we start with the fact that the system can be represented by $1+W$ transfer functions for zone *thermal load* response and $1+W$ transfer functions for zone *temperature* response where W is the number of walls exposed to different driving temperatures or sol-air conditions. In the time (z-transform) domain we have:

[3] and an established convention (Stephenson 1971, Walton 1983)

$$G_{Q/Tz}(z) = \frac{q_z}{T_z} = \frac{B_0^n(\theta_z)}{B_0^n(\phi_z)} \qquad (9) \qquad G_{Tz/Q}(z) = \frac{T_z}{q_z} = \frac{B_0^n(\phi_z)}{B_0^n(\theta_z)} \qquad (10)$$

$$G_{Q/Tw}(z) = \frac{q_z}{T_w} = \frac{B_0^n(\theta_w)}{B_0^n(\phi_z)} \qquad (11) \qquad G_{Tz/Tw}(z) = \frac{T_z}{T_w} = \frac{B_0^n(\theta_w)}{B_0^n(\theta_z)} \qquad (12)$$

The roots of the denominators correspond to the eigenvalues, λ_k, and thus to the time constants, τ_k, of any system (CTF or iCTF) of interest as follows:

$$\ell n\left(\frac{1}{r_k}\right) = \lambda_k \Delta t \qquad (13)$$

equivalently

$$\ell n(r_k) = \frac{\Delta t}{\tau_k} \qquad (14)$$

The polynomial form may be reconstructed from its roots by Pascal's triangle to satisfy:

$$\prod_{k=1}^{n}(r_k - B) = \frac{1}{x_n}\sum_{k=0}^{n} x_k B^k \qquad (15)^4$$

For diffusion processes, the characteristic frequencies, or poles, λ_k, must be real and negative, corresponding to time constants, $\tau_k = -1/\lambda_k$, that are real and positive. Note that the roots of x (θ_z or ϕ_z) are monotonic in both τ and λ. It has been shown (Hittle and Bishop 1983) that zone flux and zone temperature response function pole locations must alternate along the real axis. That is, if the roots of θ_z, $[\lambda_{\theta k}]$, and the roots of ϕ_z, $[\lambda_{\phi k}]$, are each presented as ordered sets, then together they satisfy:

$$\lambda_{\theta 1} < \lambda_{\phi 1} < \lambda_{\theta 2} < \lambda_{\phi 2} < \cdots < \lambda_{\theta n} < \lambda_{\phi n} < 0 \qquad (16)$$

Unless we constrain coefficients of the transfer function denominators, the parameter estimation process is likely to return one or more thermodynamically infeasible poles.

Model Identification. There are several ways to estimate the parameters of (6) and (8), both apparently unfettered discrete linear models, given noisy data. There are a number of different ways, in terms of problem formulation, to account for observational noise and model constraints (Golub 1980, Wunsch 2003). Three formulations relevant to zone thermal response modeling are developed below.

Model Identification with U-value Constraints. The discrete-time, linear, time-invariant simulation model (5) can, in principle, be obtained from observations of thermal response under a range of (zone heat rate and sol-air temperature) excitations by applying linear least squares.

However, unconstrained least-squares analysis minimizes the model-observation deviations without regard for the thermodynamic constraints previously noted. Moreover, the inverted model typically is very inaccurate under certain transients and it would be desirable not to have to maintain two incompatible models.

We therefore normalize temperatures by subtracting current zone (one sol-air) temperature from all other current and lagged temperatures. This eliminates the current zone (or sol-air) temperature term, reduces the order of the least squares problem by one, and results in a solution that satisfies the U-value constraint.

---

[4]Multiplying through by x_n gives $x_n\prod_{k=1}^{n}(r_k - B) = \sum_{k=0}^{n} x_k B^k$, which has cross terms but is more useful in practice; note that the vector (6) and scalar backshift operators are related by $B^{k:k}(x,y) = x_k B^k(y) = x_k y(t-k)$

Model Identification with Constrained Poles. The constraints represented by eqns (14-16) cannot be implemented in such a simple manner. However, a formulation that fits a standard bounded non-linear search model is possible. From eqns 14 and 16 we can write the constraint as

$$r_{\theta 1} > r_{\phi 1} > r_{\theta 2} > r_{\phi 2} > \cdots > r_{\theta n} > r_{\phi n} \tag{17}$$

We can now define a new search vector, $\boldsymbol{x}$, each element of which has simple bounds, $0 < x < 1$ from which the roots can be evaluated recursively:

$$\begin{aligned}
r_{\theta 1} &= x_{\theta 1} & 0 < x_{\theta 1} < 1 \\
r_{\phi 1} &= x_{\phi 1}\, r_{\theta 1} & 0 < x_{\phi 2} < 1 \\
r_{\theta 2} &= x_{\theta 2}\, r_{\phi 1} & 0 < x_{\theta 2} < 1 \\
r_{\phi 2} &= x_{\phi 2}\, r_{\theta 2} & 0 < x_{\phi 2} < 1 \\
&\vdots & \vdots \\
r_{\theta n} &= x_{\theta n}\, r_{\phi n-1} & 0 < x_{\theta n} < 1 \\
r_{\phi n} &= x_{\phi n}\, r_{\theta n} & 0 < x_{\phi n} < 1
\end{aligned} \tag{18}$$

There are two useful properties—besides reducing the multi-variable constraints (16) to a set of simple bounds on individual variables (18)—that arise from this formulation. First, we note that $x_{\theta 1}$ corresponds to the largest time constant and a reasonable upper limit on this time constant can almost always be estimated. Second, minimum separation of time constants can be imposed by using a lower bound that is greater than zero or an upper bound that is less than one.

Without pole constraints, least squares identification will typically (Seem 1985) return parameters that correspond to an unstable model (simulation grows without bound), complex or imaginary eigenvalues (resonance) or negative time constants (noncausality). None of these three properties is possible in a diffusion system. Identification with pole-constraints is not tested here; instead, we use violation of the pole constraints as a diagnostic to assess a hybrid identification scheme.

Model Identification with Uncertainty. Evaluation of thermal response is needed in both of the two modes, (6) and (8). Ordinary least squares (OLS) analysis minimizes the Euclidean norm of observation residuals. Thus if OLS is applied to heat rate observations in order to identify a CTF model (6), temperatures forecast by the corresponding iCTF model (8) are likely to have a relatively large error norm. Conversely, if OLS is applied to zone temperature observations in order to identify an iCTF model, heat rates forecast by the corresponding CTF model are likely to have a relatively large error norm. Total least squares (Golub 1980) may provide the better model parameter estimates given that reasonable estimates of the uncertainties are available for all the variables. A hybrid method that uses non-linear least squares to strike a balance between the heat rate and temperature residual error norms appears, based on this one experiment, to give good results.

Test Rooms. Two large rooms, shown in Figure 5, were instrumented to provide training and testing data with which the model and identification methods could be evaluated.

The thermal regime in a room is complex. Surface temperatures are constantly changing under the influence of conduction (most strongly in windows and exterior walls) convection and radiation. Convection may at times be dominated by buoyant forces, fan-induced, or virtually nonexistent. Boundary layer flows develop in response to surface-bulk air temperature differences. The bulk air may be stratified (stagnant or in a simple, stable mode, such as displacement ventilation), turbulent (mixed), or even divided into two regime-layers, one mixed and one stratified.

Temperature sensors generally respond to the radiant field as well as to local air temperature and the split depends strongly on local air velocity and on sensor size and emissivity. The most troublesome measurements for advanced control are therefore often the temperatures of room air and the MRT fields. For control applications, we need a single estimate of zone air temperature and/or of occupant-perceived temperature. Also note that the air stream heat rate is directly proportional to the difference between supply and exhaust temperature. The latter is often difficult to measure in the case of plenum air returns where interior and perimeter air returning at different temperatures will mix.

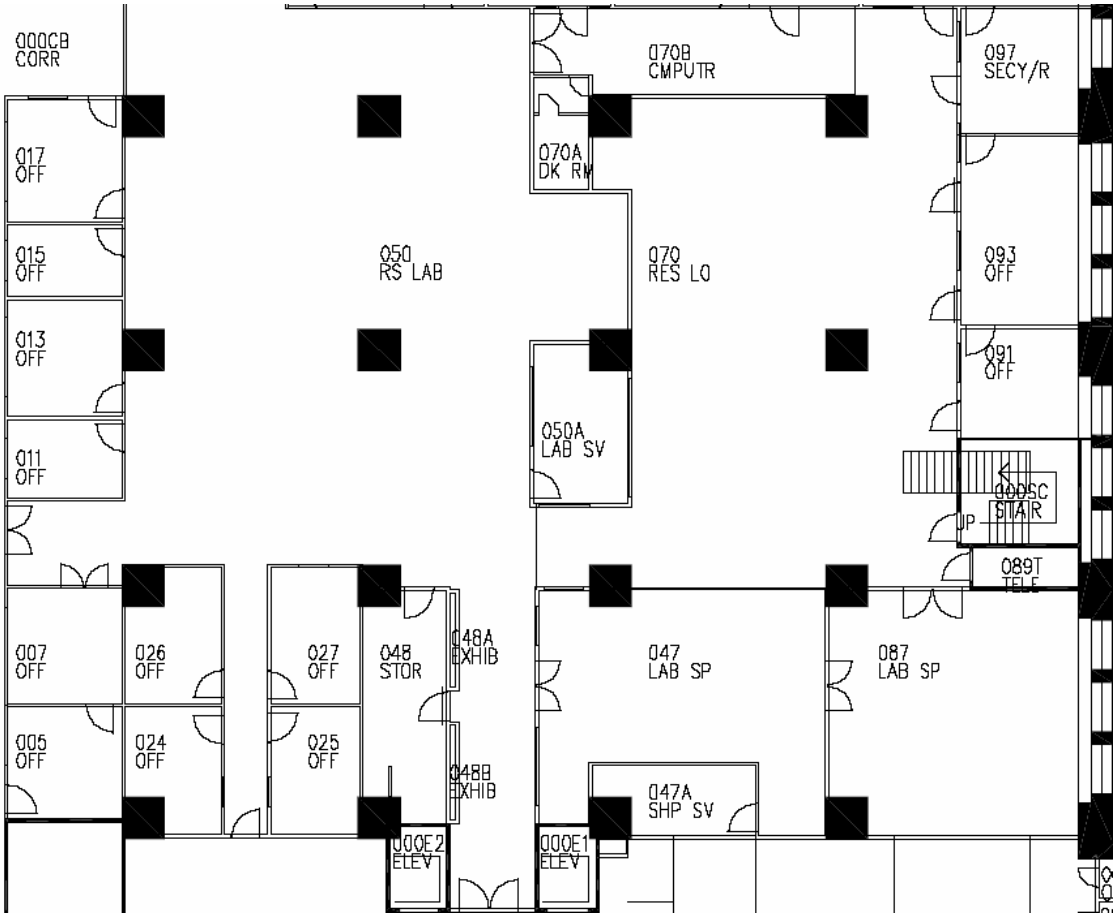

Figure 5. Main lab floor plan showing monitored zones in lower right. Room 047 is the shop and room 087 is the prototyping lab. Columns are 46”x 46” concrete.

To obtain reliable spatial average zone temperatures, as well as indications of local variations, seven to eight temperature loggers were deployed in each of the two test zones.

A two-foot diameter axial fan was set up to draw air from the main lab areas (4850 ft^2 to north and west) via a two-foot diameter supply duct, and distributes supply air to the shop (740 ft^2), to the proto lab (650 ft^2), or to both simultaneously. Zone air is allowed to return to the main lab through one or both sets of double doors.

The lighting load (ten 60W fixtures) presents a connected load of 590W = 2013 Btuh, measured by a Fluke 39 power meter. Lighting is the dominant source of internal gain in the shop. The proto-lab has computers and bench meters that add about 900 W to the 530 W connected load of

lighting. The supply air fan flow rate was measured and the corresponding thermal capacitance rate calculated to be 1620 Btuh/˚F (855 W/K). Details of wall, floor, and ceiling construction are documented in Appendix A.

Temperature loggers were deployed as indicated in Table 1 to assess local temperature variations. Additional loggers were used to monitor supply and exit air stream temperatures. The loggers sample at 2Hz, record sample averages at 5-minute intervals, and are accurate to +/-0.1˚F in the room temperature (65-80˚F) range.

Table 1. Temperature logger locations

	Logger Serial#	Location Code	Room	planLoc	elevation	Moved 20030909	Group Average
C	678777	Sh-E-1	shop	East	Floor		Shop
D	678778	Sh-N-3	shop	North	Hat		Shop
E	678779	Pl-C-2	protoLab	center	Table		protoLab
F	678780	Sh-N-1	shop	North	Floor		Shop
G	678781	Fan-In-1	fanInlet		Floor		Shop
H	678782	Pl-E-3	protoLab	East	Hat		protoLab
I	678783	Fan-In-2	fanInlet		26”AFF		
J	678784	Sh-N-2	shop	North	Table	shopOut	
K	678785	Sh-E-3	shop	East	Hat		Shop
L	678786	Pl-W-3	protoLab	West	Hat		protoLab
M	678787	Sh-S-3	shop	South	Hat		Shop
N	678788	8788	shop				Shop
O	678789	Pl-W-1	protoLab	West	Floor		protoLab
P	678790	Sh-S-2	shop	South	Table	protOut	
Q	678791	Sh-E-2	shop	East	Table	protIn	
R	678792	Sh-S-1	shop	South	Floor		Shop
S	678793	Pl-S-1	protoLab	South	Floor		protoLab
T	678794	Pl-N-1	protoLab	North	Floor		protoLab
U	678795	Pl-S-3	protoLab	South	Hat		protoLab
V	678796	Fan-out	duct				

Test Protocol Excitation and Response. The test zones experienced moderate diurnal variation in thermal conditions during the monitoring period. The existing HVAC fan coil unit serving the zones was turned off at the start of monitoring. Internal gain variations were effected by turning the lights on and off. Envelope loads resulted primarily from the temperature excursions in the main lab and outside. Selected temperature trajectories logged 6 September through 10 October are plotted in Figure 6. Figure 7 shows the same trajectories on a magnified time scale for the period of transient testing on which subsequent model identification assessment is based.

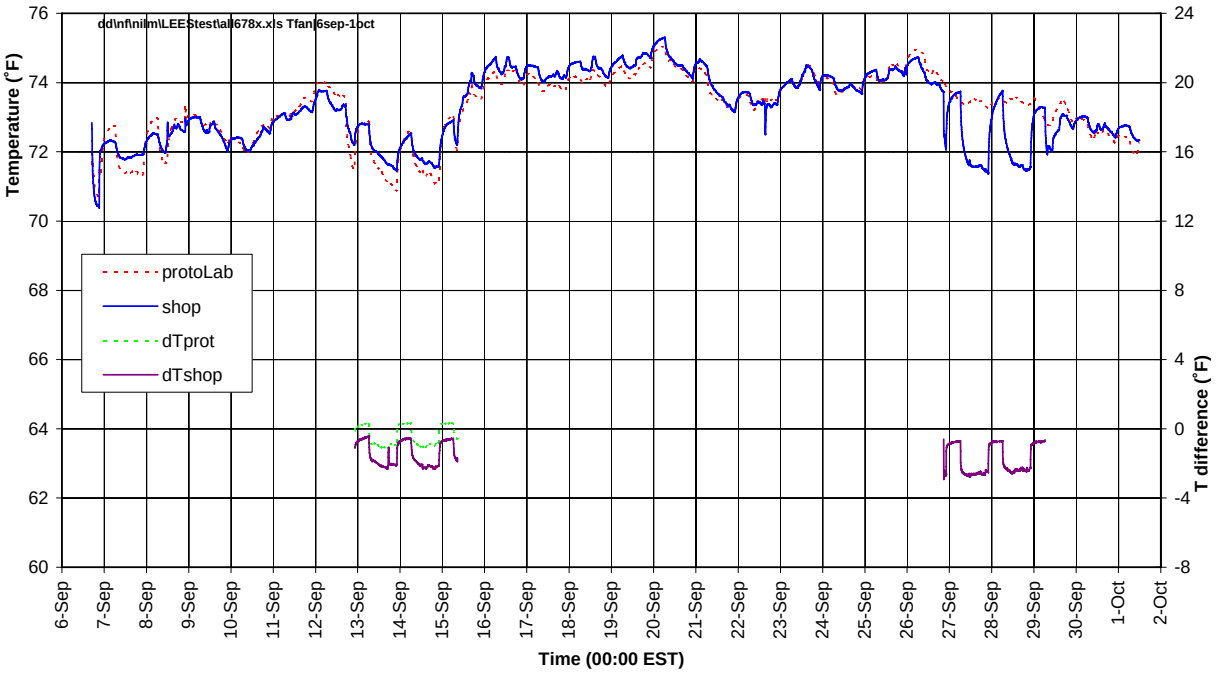

Figure 6. Temperatures measured in September and early October, 2003. Temperature differences are shown only during periods of fan operation. In the 12-15 September test air is delivered to the proto-lab and exits to the shop by the connecting door (dTprot). Air exits the shop by its west door (dTshop). In the 26-29 September test the connecting door is shut and air is delivered directly to the shop, leaving again by the west door.

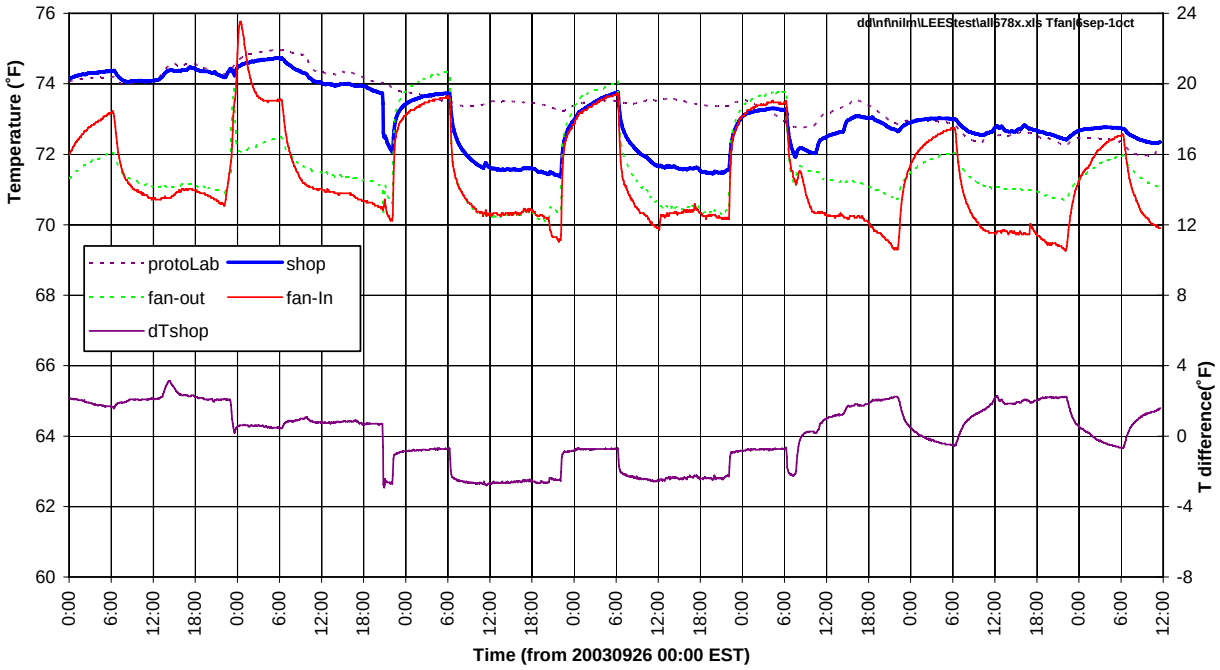

Figure 7. Temperatures measured in the test zones from 25 September to 1 October 2003.

Model Identification Results. The discrete-time, linear model developed in the Model Synthesis section will be identified and used to simulate the dynamic thermal responses of the west test

room (shop, Room 047). The parameters for a given model are determined by least-squares fit using each of the three response models: CRTF, ICRTF and hybrid. The models have identical terms but different objective functions; the identified coefficients can therefore be quite different. Each model is used to simulate, in turn, the two responses of interest: zone temperature and zone heat rate. The results are assessed by residual analysis and cross-validation.

In all three model identification cases, the exogenous temperatures and time-shifted zone temperature are converted to temperature differences by subtracting any unlagged exogenous temperature series from the other (current and lagged) temperature series. This eliminates the selected unlagged reference series term, reduces the order of the least squares problem by one, and results in a solution that satisfies the constraint on sums of temperature coefficients (7).

The first model identified for the shop is in the form of a CRTF (6) with two thermal capacitances and 15-minute time steps:

$$Q_z = B_1^n(\phi_z, Q_z) + B_0^n(\theta_w, T_w) - B_0^n(\theta_z, T_z) \qquad (6)$$

With $\phi_{z,0}$ equal minus one, the least squares solution is given in Table 2 where T is in °F and Q is in Btuh. The building load coefficient (UA) corresponding to these estimated model coefficients is 15634 Btuh/°F. The normalized width of each coefficient's confidence interval (CI) is given in Table 2.

Table 2. Coefficients, confidence intervals, and time constants from CRTF identification.

Term	Coefficient	CI/|coef|	Root	Tau(hr)
$Q_{,k-0}$	-1.0000	9.6		
$Q_{,k-1}$	1.2024	11.3	1.0205	n.a.
$Q_{,k-2}$	-0.1856	1.4	0.1819	0.34
$T_{x,k-1}$	-1383	1.2		
$T_{x,k-2}$	210	0.3		
$T_{z,k-1}$	-1351	0.7	1.3351	N.a.
$T_{z,k-2}$	2371	0.7	0.4196	0.66
$T_{x2,k-0}$	-757	0.4		
$T_{x2,k-1}$	1163	0.2		
$T_{x2,k-2}$	-2227	0.3		

Shop temperature responses produced by the 2nd-order models are compared to the measured responses in Figure 10. Cross-validation was performed by dividing the data set into a training set (27 Sep 03:30 to 28 Sep 06:30) and a testing set (28 Sep 06:30 to 29 Sep 09:30), shown on the left and right sides of Figure 10. Simulated heat input trajectories are compared to the measured trajectory in Figure 11. The training and testing sets are again shown on, respectively, the left and right sides. Residual norms are $se(Q,T_z)_{Train}$*=186, 0.138* and $se(Q,T_z)_{Test}$*=445, 0.330.*

The second model identified for the shop is in the form of an ICRTF (8) with two thermal capacitances and 15-minute time steps:

$$\theta_{z,0} T_z = B_0^n(\phi_z, Q_z) + B_0^n(\theta_w, T_w) - B_1^n(\theta_z, T_z) \qquad (8)$$

With $\phi_{z,0}$ equal minus one, the least squares solution is given in Table 3 where T is in °F and Q is in Btuh. The zone load coefficient (UA) corresponding to these estimated model coefficients is 8317 Btuh/°F.

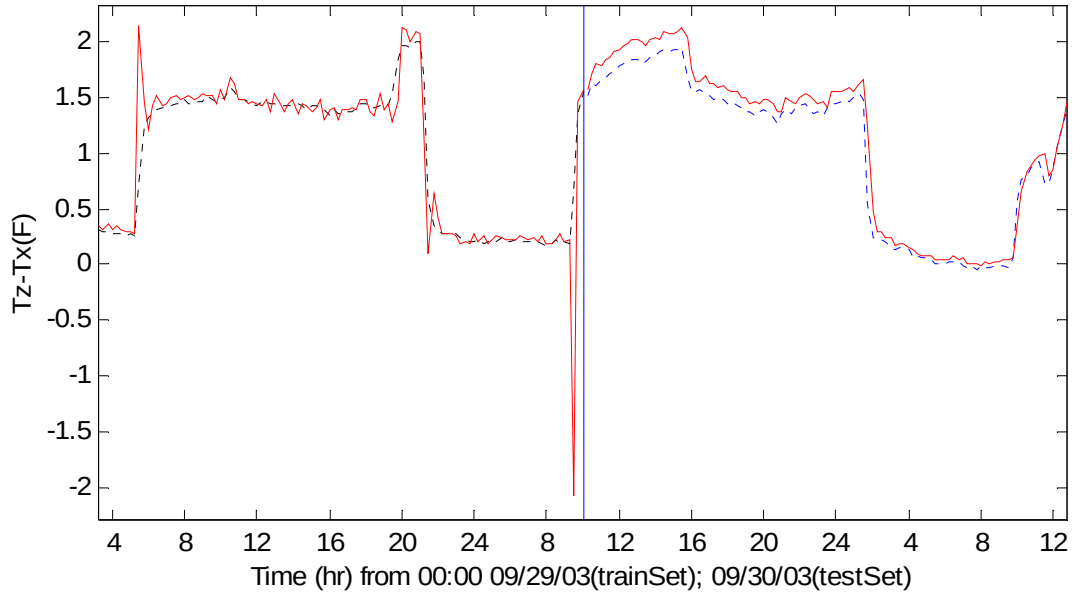

Figure 10. Forecast of temperature by inversion of identified CRTF model (solid) and observed temperature (dashed).

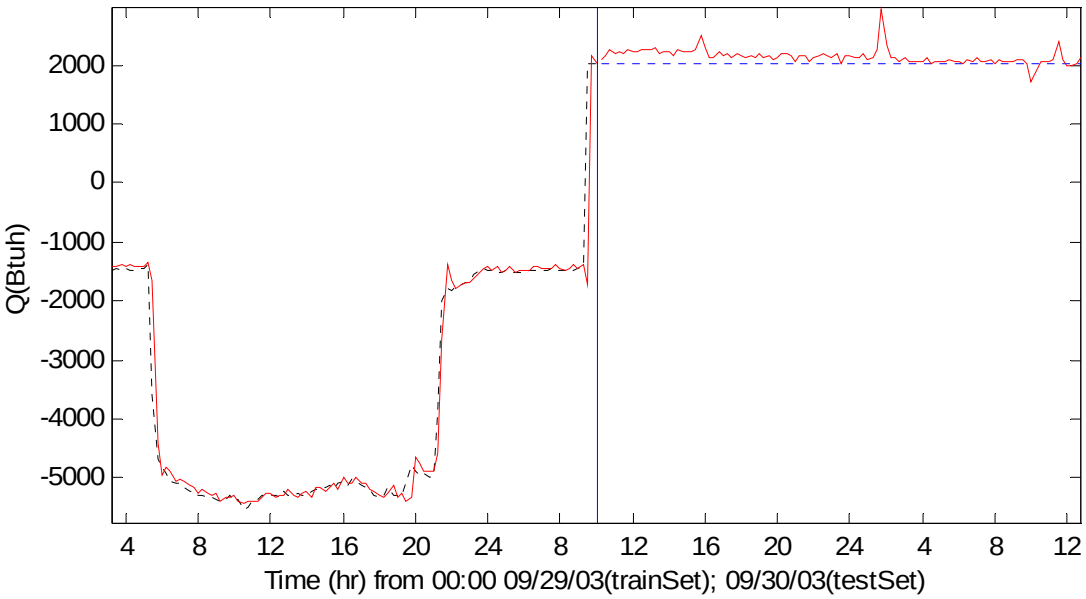

Figure 11. Forecast (solid) of heat rate by identified CRTF model and the observed heat rate (dashed).

Table 3. Coefficients, confidence intervals, and time constants; iCRTF identification.

Term	Coefficient	CI/|coef|	Root	Tau(hr)
$Q_{,k-0}$	-1.0000	0.7		
$Q_{,k-1}$	5.9271	2.7	4.5104	n.a.
$Q_{,k-2}$	-6.3899	3.5	1.4167	n.a.
$T_{x,k-1}$	-139580	14.7		
$T_{x,k-2}$	46230	5.2		
$T_{z,k-1}$	-272720	52.5	0.8887	4.88
$T_{z,k-2}$	405790	20.6	0.5992	1.12
$T_{x2,k-0}$	-145230	7.5		
$T_{x2,k-1}$	-9180	0.1		
$T_{x2,k-2}$	-4580	0.0		

Simulated shop temperature responses produced by the 2nd-order models are compared to the measured responses in Figure 12. Simulated heat input trajectories are compared to the measured trajectories in Figure 13. The training and testing sets are shown on, respectively, the left and right sides; *se(Q,T_z)$_{Train}$=11650, 0.043*; *se(Q,T_z)$_{Train}$=6328, 0.023*

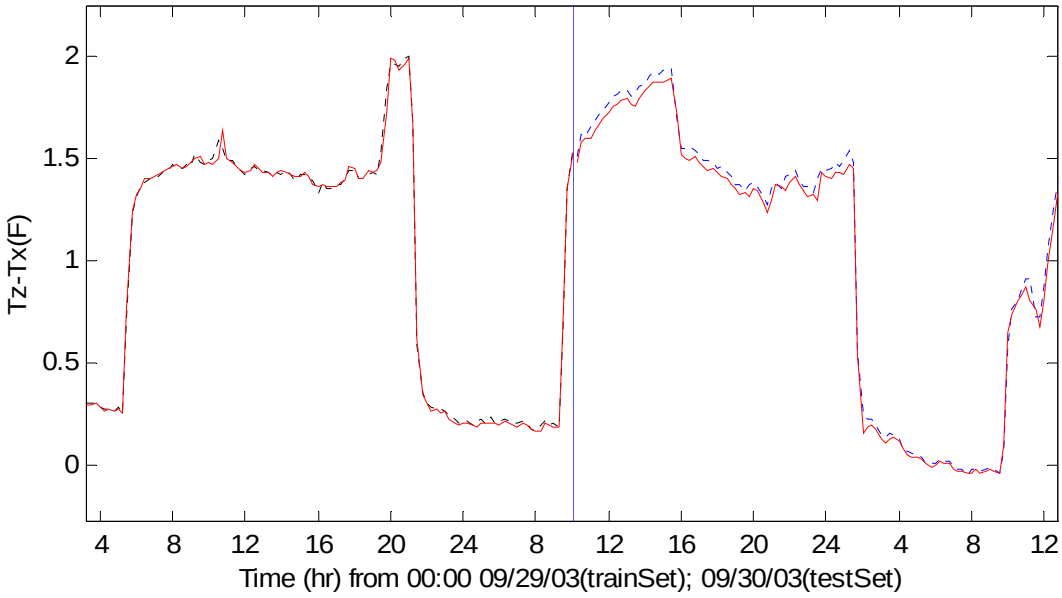

Figure 12. Forecast (solid) of temperature by identified ICRTF model and the observed heat rate (dashed).

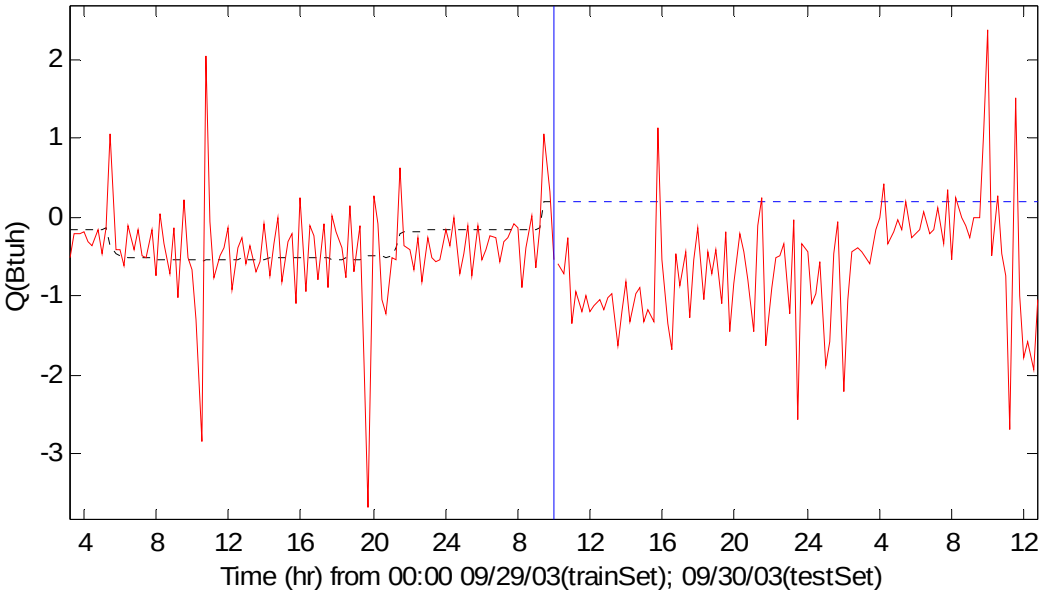

Figure 13. Forecast (solid) of heat rate by inversion of identified ICRTF model (multiply heat rate by 1000) and observed heat rate (dashed).

The third model identified for the shop is in the hybrid form (17) with two thermal capacitances and 15-minute time steps:

$$Q_z + \theta_{z,0} T_z = B_1^n(\phi_z, Q_z) + B_0^n(\theta_w, T_w) - B_1^n(\theta_z, T_z) \qquad (17)$$

With $\phi_{z,0}$ equal minus one, the least squares solution is given in Table 4 where T is in °F and Q is in Btuh. The zone load coefficient (UA) corresponding to these estimated model coefficients is *4980* Btuh/°F.

Simulated shop temperature responses produced by the 2nd-order models are compared to the measured responses in Figure 14. Simulated heat input trajectories are compared to the measured trajectories in Figure 15. The training and testing sets are shown on, respectively, the left and right sides. Training residuals norms are *se(Q,T_z)= 346, 0.0316;* testing residual norms are *se(Q,T_z)= 498, 0.0455*.

Table 4. Coefficients, confidence intervals, and time constants from hybrid identification.

Term	Coefficient	CI/|coef|	Root	Tau(hr)
$Q_{,k-0}$	-1	8.5	0.9374	8.90
$Q_{,k-1}$	1.3697	12.2	0.4323	0.69
$Q_{,k-2}$	-0.4052	2.9		
$T_{x,k-1}$	-6274	8.5		
$T_{x,k-2}$	1839	2.7	0.9636	15.52
$T_{z,k-1}$	-10956	n.a.	0.556	0.98
$T_{z,k-2}$	16650	10.8		
$T_{x2,k-0}$	-5870	3.9		
$T_{x2,k-1}$	797	0.1		
$T_{x2,k-2}$	-2310	0.2		

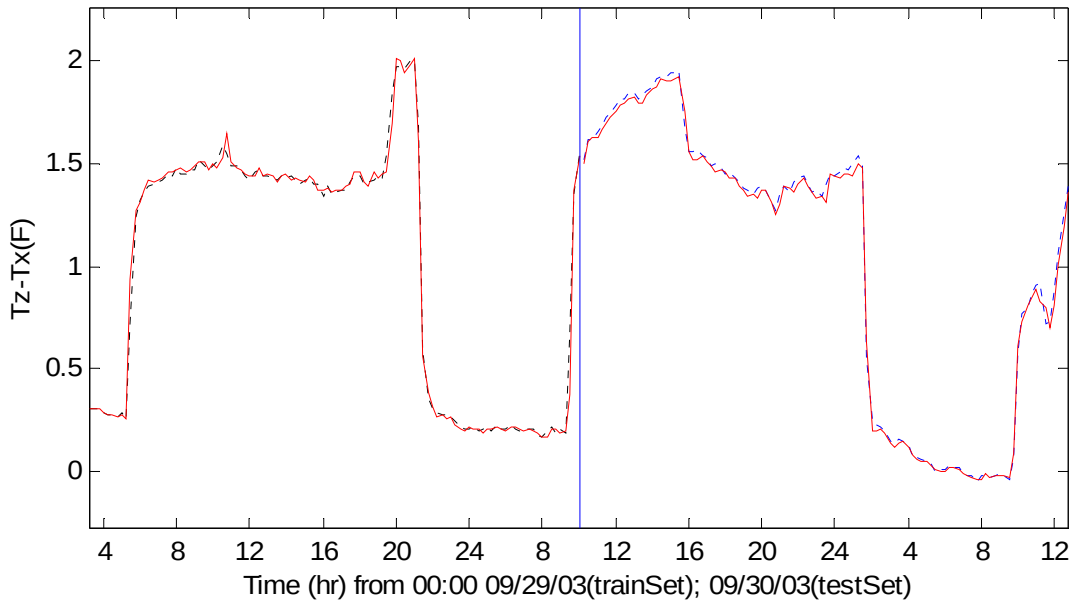

Figure 14. Temperature forecast (solid) of model identified using sum of zero-lag observations (hybrid identification). Observed temperature is the dashed line.

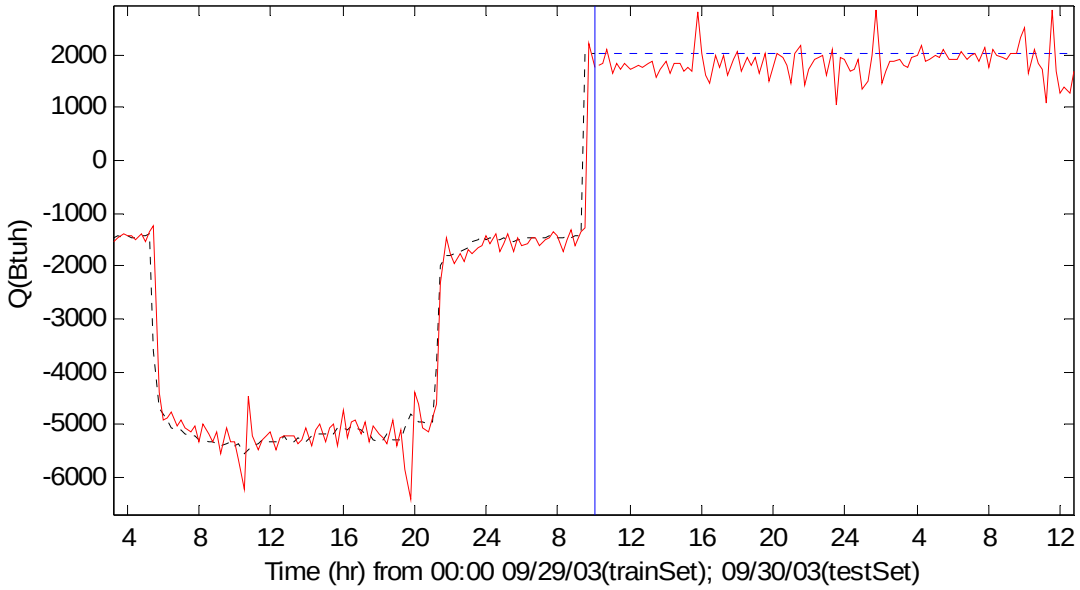

Figure 15. Heat rate forecast (solid) of model identified using weighted sum of zero-lag response observations (iCRTF responses); observed heat rate is shown by the dashed line.

Results Summary. The behavior of the three fitting procedures have been assessed in terms of the equivalent thermal parameters as well as the residual analyses.

Residual norms. The deviation of predicted from observed heat rate with CRTF model identification is on the order of 10% of rms heat rates measured during training and testing periods. The norms are 186 Btuh for training and 445 for testing. When zone temperature is simulated the norms are 0.14˚F for testing and 0.33˚F for training. These error norms are quite reasonable but the large deviations seen in Figure 10 during transients are unacceptable.

The deviation of predicted from observed zone temperature with iCRTF model identification is on the order of 5% of the rms temperature variations measured during training and testing periods. The norms are 0.02˚F for training and 0.04˚F for testing. When heat rate is simulated by the inverted model, the norms are unacceptably large: 6328 Btuh for testing and 11650 Btuh for training.

Hybrid identification gives the best of both worlds in terms of overall error norms (0.03˚F and 346 Btuh for training; 0.05˚F and 498 Btuh for testing) and deviations during transients (Figures 14-15).

UA. The zone load coefficients inferred from the three identification procedures are 15634 Btuh/˚F for the CRTF, 8317 Btuh/˚F for the iCRTF, and 4980 Btuh/˚F for the hybrid formulation.

Time constants. Two of the four time constants obtained by CRTF identification violate diffusion system constraints. A similar result is obtained with iCRTF identification. In both cases the resulting models are unstable. The hybrid identification returns time constants that are all positive real and, furthermore, obey the alternating magnitude constraint derived by Hittle and Bishop (1983).

References.

Balcomb, J.D. 1983. “Thermal network reduction,” Proc. Annual ASES Conf., Minneapolis, MN. LA-UR-83-869; also in LA-9694-MS

Barakat, S.A., 1987. “Experimental determination of the z-transfer function coefficients for houses,” ASHRAE Transactions, 93 (1) 3022.Golub, G.H. and C.F. Van Loan, 1980. “An analysis of the total least squares problem,” SIAM J. Numerical Analysis, 17(6).

Hittle, D.C. and R. Bishop, 1983. “An improved root-finding procedure for use in calculating transient heat flow through multilayered slabs,” Int’l J Heat Mass Transfer, 26(11) 1685-1693.

Seem, J.E. and Hancock E., "A Method for Characterizing the Performance of a Thermal Storage Wall from Measured Data" Thermal Performance of the Exterior Envelopes of Buildings III, ASHRAE/DOE/BTECC, (1985) Clearwater Beach, FL.

Seem, J.E., 1987. *Modeling of Heat Transfer in Buildings*, PhD Thesis, UW, Madison.

Seem, J E. and J.E. Braun, 1991. Adaptive methods for real-time forecasting of building electrical demand, *ASHRAE Trans*. 1991, vol.97, Part 1, paper# NY-910-10-3, 710-721.

Sonderegger, Robert C., "Diagnostic Tests for Determining The Thermal Response of a House," ASHRAE Transactions, AT-78-4 No. 4.

Stephenson, D.G. and Mitalas, G.P., "Calculation of Heat Conduction Transfer Functions for Multi-layer Slabs," *ASHRAE Transactions* V.77, Part II, (1971).

Subbarao, K., "Thermal Parameters for Single and Multizone Buildings and their Determination from Performance Data," Solar Energy Research Institute Report SERI/TR-253-2617 (Jan. 1985).

Walton G N. *Thermal Analysis Research Program (TARP)*. NBSIR 83-2655, March 1983.

Wunsch, C.I., 2003. *Discrete Inverse and State Estimation Problems with Fluid Flow Applications*. 15 February 2004 draft, revised chapters 1 and 2; originally published 1996 as *The Ocean Circulation Inverse Problem*, Cambridge U. Press, Cambridge.

Appendix A. Materials and Construction of Building 10 Test Zones:.
The floor is concrete cast in place on grade, thickness unknown, finished in asphalt tile.

The shop, Room 10-047, has a floor area of 740 ft^2. The north and west walls are light partitions of standard hollow-core concrete/masonry units (CMU; 4" nominal, 3.5" actual thickness) to 56" AFF and glass to 104" above finished floor (AFF). The shop is entered by a 60 x 84" double door on its west wall.

The dividing wall between the two zones is CMU (4" nominal, 3.5" actual thickness) extending from floor to top of ceiling plenum (124"AFF). A 60 x 84" double door provides access between the two rooms. The 46" square columns at either end of the east wall extend 42" into the proto-lab space and are flush with the partition on the shop side.

The proto-lab, Room 10-087, has a floor area of 650 ft^2. The north wall is CMU (4" nominal, 3.5" actual thickness) extending from floor to ceiling (104"AFF) with a72 x 90" double door leading to the main lab. The east wall is 8" CMU to 56" AFF and metal frame windows, facing a parking lot, to the ceiling. The window units are 64 x 88" in 16" deep concrete mullions. The top mullion is 12" high and the columns between windows are 16" wide. The 46" square columns at either end of the east wall extend 29" into the space and 16" to the outside.

The south wall of both rooms appears to be of 8" CMU; same for the shop pantry wall. The pantry is entered by a 36x 90" door.

The shop ceiling is suspended acoustic tile (SAT) in a standard 24" grid, suspended 104" AFF. The proto-lab ceiling is suspended acoustic tile (SAT) in a standard 24" grid, suspended 94.5" AFF.

Appendix B. Model Canonical Forms.

Inversion, i.e., transformation between the CRTF (6) and iCRTF (8) models is a frequent and sometimes confusing operation (see simulation code listings in Appendices E and F). A set of four canonical forms have been found to be useful. The coefficient index notation (i:n) indicates the range of a coefficient vector's association with past samples. Thus, for example:

$\phi_z(0{:}n) = [\phi_{z,0}\ \phi_{z,1} \cdots \phi_{z,n}]$ $\quad\quad$ $\phi_z(1{:}n) = [\phi_{z,1}\ \phi_{z,2} \cdots \phi_{z,n}]$

$\theta_z(0{:}n) = [\theta_{z,0}\ \theta_{z,1} \cdots \theta_{z,n}]$ $\quad\quad$ $\theta_z(1{:}n) = [\theta_{z,1}\ \theta_{z,2} \cdots \theta_{z,n}]$

$\theta_w(0{:}n) = [\theta_{w,0}\ \theta_{w,1} \cdots \theta_{w,n}]$ $\quad\quad$ $\theta_w(1{:}n) = [\theta_{w,1}\ \theta_{w,2} \cdots \theta_{w,n}]$

The range of lags may also be expressed in operator notation by the backshift operator with the range expression appended as a superscript:

$B^{i:n}(\phi) = [\phi_i\ \phi_{i+1} \cdots \phi_n]$

$B^{i:n}(\theta) = [\theta_i\ \theta_{i+1} \cdots \theta_n]$

$B^{i:n}(q) = [\ q(t\text{-}i)\ q(t\text{-}i\text{-}1) \cdots q(t\text{-}n)]$

$B^{i:n}(T) = [T(t\text{-}i)\ T(t\text{-}i\text{-}1) \cdots T(t\text{-}n)]$

Equations (6) and (8) were developed for the simplest case involving just one exogenous temperature, e.g. a wall model. Here we extend the subscript used with temperature coefficients to indicate which temperature is meant: z refers to the inside wall surface or zone temperature and $w = 1{:}W$ refers to exogenous temperatures or sol-air temperatures 1 through W.

Note that the zero-lag term of the first exogenous temperature is absent from all of the forms. Its value is given by the u-value constraint (7).

CRTF form

$x_c = [Q(t\text{-}k,\ k=1{:}n)\ T_z(t\text{-}k,\ k=0{:}n)\ T_1(t\text{-}k,\ k=1{:}n)\ T_2(t\text{-}k,\ k=0{:}n) \cdots T_W(t\text{-}k,\ k=0{:}n)]$

$b_c = [\phi(1{:}n)\ \theta_z(0{:}n)\ \theta_1(1{:}n)\ \theta_2(0{:}n) \cdots \theta_w(0{:}n)]$ with $\phi(0) = -1$ implicit

$Q = x_c\ b_c$

iCRTF form

$x_i = [Q(t\text{-}k,\ k=0{:}n)\ T_z(t\text{-}k,\ k=1{:}n)\ T_1(t\text{-}k,\ k=1{:}n)\ T_2(t\text{-}k,\ k=0{:}n) \cdots T_W(t\text{-}k,\ k=0{:}n)]$

$b_i = [\phi(0{:}n)\ \theta_z(1{:}n)\ \theta_1(1{:}n)\ \theta_2(0{:}n), \cdots, \theta_w(0{:}n)]$ with $\theta_z(0) = -1$ implicit

$T_z = x_i\ b_i$

Un-normalized complete form

$x_u = [Q(t\text{-}k,\ k=0{:}n)\ T_z(t\text{-}k,\ k=0{:}n)\ T_1(t\text{-}k,\ k=1{:}n)\ T_2(t\text{-}k,\ k=0{:}n) \cdots T_W(t\text{-}k,\ k=0{:}n)]$

$b_u = [\phi(0{:}n)\ \theta_z(0{:}n)\ \theta_1(1{:}n)\ \theta_2(0{:}n), \cdots, \theta_w(0{:}n)]$ with no particular scaling imposed

$0 = x_u\ b_u$

Q-normalized complete form

$x_n = [Q(t\text{-}k,\ k=0{:}n)\ T_z(t\text{-}k,\ k=0{:}n)\ T_1(t\text{-}k,\ k=1{:}n)\ T_2(t\text{-}k,\ k=0{:}n) \cdots T_W(t\text{-}k,\ k=0{:}n)]$

$b_n = [\phi(0{:}n)\ \theta_z(0{:}n)\ \theta_1(1{:}n)\ \theta_2(0{:}n), \cdots, \theta_w(0{:}n)]$ with $\phi(0) = -1$ explicit.

$0 = x_n\ b_n$

It is apparent that a T_z-normalized form can also be immediately defined. We prefer in practice to use an iCRTF form obtained, on the fly, from the CRTF form:

…or from either one of the complete forms:

Note: we insert the t-ratio from the residual error norm when converting from CRTF or iCRTF canonical form to either of the complete forms (see function rootstu(bs,n)).

Appendix C. Model Identification by Ordinary and Hybrid Least Squares

```
>>agglees9('lees1oct',1,2000,890,919,1210,600,[2],[3]);
function b=aggLEES9(fnam,fLtg,mcp,jhead,jtail,khead,ktail,nlist,malist);
%train/test sets from fnam between head and tail line#s;
%fit model orders in nlist, e.g. [0 1 2 3] and
%time-aggregations in malist, e.g. [1 2 3 6]
%fLtg, mcp for problem specific heat rate calcs
%aggLEES9.m 20031218pra use canonical order within x,b
data=load([fnam '.txt']); size(data);
%1:8 lotusdate TfanIn1 TfanIn2 TfanOut Tprot Tshop dTube dTshop (degF)
time.nXTicks=15; time=timecalc(data,jhead,khead,time);
for ma=malist;
   %---------problem specific heat rate calcs---------
   q=data(:,1)>=37890.86806 & data(:,1)<37892.26389; %ltgOff & fanOn
   %q=2390*(1-data(:,1))+2581*data(:,1).*data(:,8); %Qltg+G*dTshop
   Qltg=fLtg*590*3600/1055;%fLtg*[W]*[W2Btuh]=fLtg*2013Btuh
   q=Qltg*(1-q)+mcp*q.*data(:,8); %Qltg+G*dTshop
   %---------end problem-specific heat rate calcs---------

   [jq,jz,jx,jx2,jn]=laggedTs([q data(:,6) data(:,3) data(:,5)],ma,jhead,jtail,nlist);
   [kq,kz,kx,kx2,kn]=laggedTs([q data(:,6) data(:,3) data(:,5)],ma,khead,ktail,nlist);
   time=timacalc(ma,jn,kn,time);
   plotqtzi([1,2],jq,jz,jn,kq,kz,kn,time,'iCRTFid')
   plotqtzi([3,4],jq,jz,jn,kq,kz,kn,time,'CRTFid')
   plotqtzi([5,6],jq,jz,jn,kq,kz,kn,time,'QTzid')
   uscale=1.
   for n=nlist;%model order (number of poles)
      np=n+1
      xj=[jq(:,1:np) jz(:,1:np) jx(:,2:np) jx2(:,1:np)];
      xk=[kq(:,1:np) kz(:,1:np) kx(:,2:np) kx2(:,1:np)];

      y=jz(:,1);
      x=[uscale*jq(:,1:np) jz(:,2:np) jx(:,2:np) jx2(:,1:np)];%Tz-response

      disp('iCRTF LScovWgt')
      alph2=0;
      Rbb=alph2*eye(length(x(1,:)));Rnn=eye(length(x(:,1)));
      [b,bse]=lscovwgt(x,Rbb,y,Rnn);
      bs=bi2bns(b,bse,n,x,y)
      bs2str(bs,n);
      u=rootstu(bs(:,1),bs(:,2),n)
      seqtz=plotcrtf([1,2],xj,n,bs,0);
      seqtz=plotcrtf([1,2],xk,n,bs,jn);

      if 0
      disp('iCRTF lscov')
      [b,bse]=lscov(x,jz(:,1),eye(length(x(:,1))));
      bs=bi2bns(b,bse,n,x,jz(:,1))
      %bs=[b bse abs(b./bse)];
      bs2str(bs,n);
      u=rootstu(bs(:,1),bs(:,2),n)

      disp('iCRTF LSVDcov')
      [b,bse]=lsvdcov(x,jz(:,1),eye(length(x(:,1))));
      bs=bi2bns(b,bse,n,x,jz(:,1))
      bs2str(bs,n);
      u=rootstu(bs(:,1),bs(:,2),n)
      end

      y=uscale*jq(:,1);
      x=[uscale*jq(:,2:np) jz(:,1:np) jx(:,2:np) jx2(:,1:np)];%q-response
      disp('CRTF LScovWgt')
      alph2=0;
      Rbb=alph2*eye(length(x(1,:)));Rnn=eye(length(x(:,1)));
      [b,bse]=lscovwgt(x,Rbb,y,Rnn);
      bs=bc2bns(b,bse,n,x,y)
      bs2str(bs,n);
      u=rootstu(bs(:,1),bs(:,2),n)
      seqtz=plotcrtf([3,4],xj,n,bs,0);
      seqtz=plotcrtf([3,4],xk,n,bs,jn);
```

```
        bz=-bs(np+1)
       if bz<200; bz=1000;    end
      xx=[jq(:,2:np) jz(:,2:np) jx(:,2:np) jx2(:,1:np)];
      bz=fminbnd('lsqz',bz,50*bz,[],xx,y,jz(:,1))
      [b,bse]=lscov(xx,y+bz*jz(:,1),eye(length(y)));%b s/b unchngd
      b=[b(1:n);-bz;b(np:length(b))];
      bse=[bse(1:n);0;bse(np:length(bse))];
      bs=bc2bns(b,bse,n,x,y)
      bs2str(bs,n);
      u=rootstu(bs(:,1),bs(:,2),n)
      seqtz=plotcrtf([5,6],xj,n,bs,0);
      seqtz=plotcrtf([5,6],xk,n,bs,jn);
   end;%n
end;%ma
for i=1:6;figure(i);v=axy(.5); plot([jn jn],[v(3),v(4)]);hold off;end
return;%-----------------------------------------------------------------

function v=axy(f)
vi=axis; axis('tight');
v=axis;
vi=(1-f)*v+f*vi; %tighter y limits
v=[v(1) v(2) vi(3) vi(4)]; %x-fullyTight; y-partlyTight
axis(v); %apply
return;%-----------------------------------------------------------------
function e=lsqz(bz,x,yq,yz)
%cost fcn name passed to fminbnd('lsqz',bzl,bzu,options,x,y,kz(:,1))
%x is the CRTF data matrix with Tzone column removed
%yq=qzone is the vector observations; u=theta0 associated with yz=Tzone
y=yq+bz*yz; cov=eye(length(y)); one=diag(cov);
rmsq=norm(yq-mean(yq)*one);
rmsz=norm(yz-mean(yz)*one);
b=lscov(x,y,cov);
ye=x*b;
e=norm(yq-ye+bz*yz)/rmsq +norm(yz-(ye-yq)/bz)/rmsz;
return;%-----------------------------------------------------------------
function [kq,kz,kx,kx2,kn]=laggedTs(data,ma,khead,ktail,nlist)
%function [kq,kx,kz,kx2]=laggedTs(data,clist,ma,khead,ktail,nlist)
nn=max(nlist);
m=fix((length(data(:,1))-khead-ktail-ma*nn)/ma)
mm=m*ma
k=khead+1; kf=khead+mm+ma*nn;
q=decima(data(k:kf,1),ma); %Qltg+G*dTshop
tz=decima(data(khead+1:khead+mm+ma*nn, 2),ma);
fprintf(1,'%21s Tshop staircase\n',datestr(now));
wp=length(data(1,:)); %future multiple exogenous temperatures
tx=decima(data(khead+1:khead+mm+ma*nn, 3),ma); %Tmain=TfanIn2
tx2=decima(data(khead+1:khead+mm+ma*nn, 4),ma); %Tprot
%NOTE len(tx,tz,q) = len(kx,kz,kq)+nn
kq= q(1+nn:m+nn);
kx=tx(1+nn:m+nn); kx0=kx;
kz=tz(1+nn:m+nn)-kx0;
kx2=tx2(1+nn:m+nn)-kx0;
kn=length(kq);
for n=1:nn
   k=nn-n;
   kq=[kq q(1+k:m+k)];
   kz=[kz tz(1+k:m+k)-kx0];
   kx=[kx tx(1+k:m+k)-kx0];
   kx2=[kx2 tx2(1+k:m+k)-kx0];
end
return;%-----------------------------------------------------------------
function t=timecalc(data,jhead,khead,t);
t.StepHr=[1 2 3 4 6 8 12 18 24 36 48];%allow these axis increments
%following round-off correction works only if 1/t.StepHr is an integer
t.StepHr=1.0/round(1/(24*(data(2,1)-data(1,1))));
t.Startj=data(jhead,1)+1900*365.25-13;%convert from lotusdate to a.d.
t.StartjRem=t.Startj-floor(t.Startj);
t.Startk=data(khead,1)+1900*365.25-13;%convert from lotusdate to a.d.
t.StartkRem=t.Startk-floor(t.Startk);
%tickStartHr=ceil(24*t.StartRem)
t.StartjStr=datestr(t.Startj,2);%convert to datestring
```

```
t.StartkStr=datestr(t.Startk,2);%convert to datestring
return;%-------------------------------------------------------------
function t=timacalc(ma,jn,kn,t);
   t.StepAggHr=ma*t.StepHr;
   %following works only if 1/tAggHr is an integer
   t.StepXTickHr=ceil((jn+kn)*t.StepAggHr/t.nXTicks);
   t.StepsPerTick=t.StepXTickHr/t.StepAggHr;
   t.StartjTickHr=t.StepXTickHr*ceil(24*t.StartjRem/t.StepXTickHr);
   t.Startjmndex=round(rem(t.StartjTickHr,24)/t.StepAggHr);
   t.Startjindex=round(t.StartjTickHr-24*(t.StartjRem))/t.StepAggHr;
   t.StartkTickHr=t.StepXTickHr*ceil(24*t.StartkRem/t.StepXTickHr);
   t.Startkmndex=round(rem(t.StartkTickHr,24)/t.StepAggHr);
   t.Startkindex=round(t.StartkTickHr-24*(t.StartkRem))/t.StepAggHr;
   t.BaseXLabel=['Time (hr) from 00:00 ' t.StartjStr '(trainSet); ' t.StartkStr
'(testSet)'];
return;%-------------------------------------------------------------
function plotqtzi(fig,jq,jz,jn,kq,kz,kn,t,ti)
%could pass jn,kn as t.jn,t.jn
   figure(fig(1)); plot(jq(:,1),'k:'); title(ti);
   s=t.StepsPerTick; set(gca,'XTick',[[t.Startjindex:s:jn] [jn+t.Startkindex:s:jn+kn]]);
   set(gca,'XTickLabel',round(t.StepAggHr*[[t.Startjmndex:s:jn] [t.Startkmndex:s:kn]]));
   xlabel(t.BaseXLabel); ylabel('Q(Btuh)');hold on
   plot([jn+1:jn+kn],kq(:,1),'b:');
   figure(fig(2)); plot(jz(:,1),'k:'); title(ti);
   set(gca,'XTick',[[t.Startjindex:s:jn] [jn+t.Startkindex:s:jn+kn]]);
   set(gca,'XTickLabel',round(t.StepAggHr*[[t.Startjmndex:s:jn] [t.Startkmndex:s:kn]]));
   %set(gca,'XTick',[[1:s:jn] [jn+1:s:jn+kn]]);
   xlabel(t.BaseXLabel); ylabel('Tz-Tx(F)');hold on
   plot([jn+1:jn+kn],kz(:,1),'b:');
return;%-------------------------------------------------------------
function seqtz=plotcrtf(fig,x,n,bs,tb)
%usu figure(fig(1:2)) already exist w/axes, titles, plot of observed response
c='bgrycmk';%tb=base time index; for aggLEES.m tb(train)=0, tb(test)=jn
np=n+1;
b=bs(:,1);
  %CRTF (Qz response)
  y=x(:,1)-(x*b)/b(1);
  figure(fig(1));plot([tb+1:tb+length(y)],y,c(np))
  y=x(:,1)-y; sse=y'*y;%reuse y for r
  seqtz(1)=sqrt(sse/(length(y)+1-3*np));

  %ICRTF (Tz as response)
  iz=np+1; %bz=b(iz);
  y=x(:,iz)-x*b/b(iz);
  figure(fig(2));plot([tb+1:tb+length(y)],y,c(np))
  y=x(:,iz)-y; sse=y'*y;
  seqtz(2)=sqrt(sse/(length(y)+1-3*np));
  sprintf('se(q,tz)= %0.5g %0.5g',seqtz)
return;%-------------------------------------------------------------
function b=rootstu(b,bse,n)
%given un-normalized complete b, print useful stuff; later: return log obj
%assumes b in un-normalized complete form: q[0:n] Tz[0:n] Tx[1:n] Tx2[0:n]
np=n+1; W=(length(b)+1)/np - 2
%in either complete canonical form: u=sum(ce)/sum(de)
utot=sum(b(np+1:np+np))/(sum(b(1:np)))
for i=2:W;j=np*(i+1); u(i)=sum(b(j:j+n))/(sum(b(1:np))); end
u(1)=utot-sum(u(2:W))
seqtz=[bse(1) bse(np+np)]
%be=b(np:np+n-1); %insert be(0)?
ce=b(np+1:2*np);
rc=roots(ce)
;nrc=sum(imag(rc)==0);%abs(rc)==-rc
de=b(1:np);
rd=roots([de])
;nrd=sum(imag(rd)==0);%isreal(rd)
return;%-------------------------------------------------------------
function b=bi2bn(bi,n)
np=n+1;
b=[bi(1:np,:) 0*bi(1,:) bi(np+1:length(bi),:)];%insert null row
b=-b/bi(1);
return;%-------------------------------------------------------------
```

```
function bs=bc2bns(b,be,n,x,y)
np=n+1;%[b,bse] convert CRTF model id result to normalized canonical form
r=y-x*b; %compute std error of fit
se=sqrt((r'*r)/(length(y)-length(b)));

%move observation [coeff,se] to RHS
bs(:,1)=[-1; b(1:np+n); b(np+np:length(b))];
bs(:,2)=[sqrt((r'*r)/(y'*y)); be(1:np+n); be(np+np:length(b))];
 %be(1:np+n); se; be(np+np:length(b))];

%normalize
%bs=-bs/b(1);
%compute t-ratios
%bs(:,3)=abs(bs(:,1)./bs(:,2))
return;%-----------------------------------------------------------------

function bs=bi2bns(b,be,n,x,y)
np=n+1;%[b,bse] convert iCRTF model id result to normalized canonical form
r=y-x*b; %compute std error of fit
se=sqrt((r'*r)/(length(y)-length(b)));
%THIS IS se(data); NEED TO CONVERT TO se(coeff) using expression for CV?

%move observation [coeff,se] to RHS
bs(:,1)=[b(1:np); -1; b(np+1:length(b))];
bs(:,2)=[be(1:np); sqrt((r'*r)/(y'*y)); be(np+1:length(b))]; %s/b se(coeff)! ! !
 %be(1:np); se; be(np+1:length(b))];

%normalize
bs=-bs/b(1);
%compute t-ratios
%bs(:,3)=abs(bs(:,1)./bs(:,2))
return;%-----------------------------------------------------------------

function bstr=bs2str(bs,n)
np=n+1;%pretty-print the (complete) model [coeff,se] vectors and t-ratios
%scale by u-value?
t=abs(bs(:,1)./bs(:,2))
xt=[np+1:length(t)];

if 0;
   blanco=char([0*t+32]);
   bstr=[num2str(bs(:,1)),blanco,num2str(bs(:,2)),blanco,num2str(t)];
else
   disp([bs(1:np,:),t(1:np)])
   disp([bs(xt,:), t(xt)])
end
return
blanco=char([0*t(1:np)+32]);
bstr=[num2str(bs(1:np,1)),blanco,num2str(bs(1:np,2)),blanco,num2str(t(1:np))]
xt=[np+1:length(t)];
blanco=char([0*t(xt)+32]);
bstr=[[num2str(bs(xt,1)),blanco,num2str(bs(xt,2)),blanco,num2str(t(xt))]];
return;%-----------------------------------------------------------------

function [b,be]=lsvdcov(x,y,cov)
%Numercal Recipes Ftn (1992) 15.4.3-5 and 15.4.18-19
%corrected per Wunsch,C.I., Ch 2.5, 2003 rev of The Ocean Circulation Inverse Problem
(1996)
m=length(x(1,:))
yb=cov\y;%yb=y/diag(cov);%for diagonal cov
%[u,w,v]=svd(cov\x,0);%svd([x(i,j)/sig(i)]);for diag
[u,w,v]=svd(x,0);%svd([x(i,j)/sig(i)]);for diag
s=diag(w);
b=0*x(1,:)';be=b; k=0; tol=sqrt(m)*s(1)*4e-15
while s(k+1)>tol;
   k=k+1;
   if k==m;
      break;
   end
end
%below tol tests c/b avoided by using k as upper summation limit
```

```
for i=1:m;
   if s(i)>tol; b=b+(u(:,i)'*yb)*v(:,i)/s(i); end
   for j=1:m;
      if s(j)>tol; be(i)=be(i)+(v(i,j)/s(j))^2;end
   end
   be(i)=sqrt(be(i));
end
return;%-----------------------------------------------------------------
function [b,be]=lsvdw(x,y)
%Wunsch,C.I., Ch 2.5, 2003 rev of The Ocean Circulation Inverse Problem (1996)
m=length(x(1,:))
yb=y;
[u,w,v]=svd(x,0);%svd([x(i,j)/sig(i)]);for diag
s=diag(w);overs=1./s
b=0*x(1,:)';be=b; ey=0*yb; p=0*eye(m); k=0; tol=sqrt(m)*s(1)*4e-15
while s(k+1)>tol; k=k+1; if k==m; break; end; end
for i=1:k;
   b=b+overs(i)*(u(:,i)'*yb*v(:,i));%(5.77) w/o bias part
   ey=ey+(u(:,i)'*yb)*u(:,i);;      %(5.78)
   p =p +overs(i)*overs(i)*(v(:,i)*v(:,i)');%(5.83) w/o bias part
end
ey=ey-yb;
p=diag(p)
length(yb)-m
be=sqrt(((ey'*ey)/(length(yb)-m))*p);
return;%-----------------------------------------------------------------
function [b,be]=lscovwgt(E,Rbb,y,Rnn)
%Wunsch,C.I., Ch 2.5, 2003 rev of The Ocean Circulation Inverse Problem (1996)
%eqns 2.7.19-21 with provision for Rbb=0
Rnn=inv(Rnn);
m=length(E(1,:))
if rank(Rbb)==m; p=inv(inv(Rbb)+E'*Rnn*E);
else; p=inv(E'*Rnn*E);
end
b=p*E'*Rnn*y;
ey=y-E*b;
p=diag(p);
length(y)-m
be=sqrt(((ey'*ey)/(length(y)-m))*p);
return;%-----------------------------------------------------------------

function [b,be,k]=lsvdwgt(E,Rbb,y,Rnn)
%Wunsch,C.I., Ch 2.5, 2003 rev of The Ocean Circulation Inverse Problem
%(1996);  %COULD USE Rbb,RnnCholeski factors and work in xfmd space
Rbb=inv(Rbb);Rnn=inv(Rnn);
m=length(E(1,:))
yb=y;
[u,w,v]=svd(x,0);%svd([x(i,j)/sig(i)]);for diag
s=diag(w);overs=1./s
b=0*x(1,:)';be=b; ey=0*yb; p=0*eye(m); k=0; tol=sqrt(m)*s(1)*4e-15
while s(k+1)>tol; k=k+1; if k==m; break; end; end
for i=1:k;
   b=b+overs(i)*(u(:,i)'*yb*v(:,i));%(5.77) w/o bias part
   ey=ey+(u(:,i)'*yb)*u(:,i);;      %(5.78)
   p =p +overs(i)*overs(i)*(v(:,i)*v(:,i)');%(5.83) w/o bias part
end
ey=ey-yb;
be=sqrt(((ey'*ey)/(length(yb)-m))*diag(p));
return;%-----------------------------------------------------------------

function b=tlsq(scaleq,nn,x,y)
%Golub & VanLoan, 1989, Matrix Computations, Algorithm 5.1.1
np=nn+1; n=length(x(1,:)); m=length(x(:,1));%n data; m parms
sm=[scaleq*ones(np:1);ones(n-np)];
%use nn for CRTF canonical form; np for complete form
sm=diag(sm);%x-->x*sm [3 PL]
[u,w,v]=svd([x y],0);
[diag(w) log10(diag(w))]
er=-[x y]*v*v';
b=[x+er(:,1:n)]\[y+er(:,n+1)];
return;%-----------------------------------------------------------------
```

Appendix G. Fragment of data LEES1oct.txt for input to aggLEES9.m

```
lotusdate TfanIn1 TfanIn2 TfanOut Tprot Tshop dTube dTshop
37891.14583 73.69 73.52 74.21 73.61 73.84 -0.576 -0.737
37891.14931 73.73 73.52 74.21 73.62 73.85 -0.562 -0.737
37891.15278 73.69 73.52 74.25 73.61 73.85 -0.562 -0.737
37891.15625 73.73 73.52 74.25 73.61 73.85 -0.547 -0.737
37891.15972 73.73 73.52 74.25 73.61 73.85 -0.547 -0.737
37891.16319 73.69 73.56 74.25 73.60 73.84 -0.548 -0.737
37891.16667 73.73 73.52 74.25 73.59 73.85 -0.547 -0.737
37891.17014 73.73 73.52 74.25 73.59 73.85 -0.547 -0.737
37891.17361 73.73 73.56 74.29 73.59 73.86 -0.562 -0.693
37891.17708 73.73 73.56 74.25 73.58 73.86 -0.533 -0.780
37891.18056 73.78 73.56 74.29 73.58 73.87 -0.548 -0.736
37891.18403 73.78 73.56 74.29 73.56 73.86 -0.548 -0.736
37891.1875  73.78 73.56 74.29 73.56 73.86 -0.548 -0.693
37891.19097 73.78 73.56 74.25 73.55 73.86 -0.519 -0.780
37891.19444 73.73 73.56 74.25 73.55 73.86 -0.577 -0.693
37891.19792 73.78 73.56 74.29 73.55 73.87 -0.548 -0.736
37891.20139 73.78 73.56 74.29 73.54 73.86 -0.548 -0.693
37891.20486 73.78 73.60 74.29 73.53 73.88 -0.534 -0.736
37891.20833 73.78 73.60 74.29 73.53 73.87 -0.534 -0.736
37891.21181 73.78 73.60 74.29 73.52 73.88 -0.534 -0.736
37891.21528 73.78 73.60 74.29 73.52 73.88 -0.534 -0.736
37891.21875 73.73 73.60 74.29 73.53 73.87 -0.548 -0.736
37891.22222 73.78 73.60 74.29 73.52 73.88 -0.534 -0.736
37891.22569 73.78 73.60 74.29 73.52 73.88 -0.534 -0.736
37891.22917 73.78 73.60 74.29 73.52 73.88 -0.534 -0.736
37891.23264 73.78 73.60 74.34 73.52 73.88 -0.519 -0.736
37891.23611 73.78 73.60 74.34 73.52 73.89 -0.562 -0.693
37891.23958 73.78 73.65 74.34 73.52 73.88 -0.505 -0.736
37891.24306 73.78 73.60 74.34 73.52 73.88 -0.519 -0.736
37891.24653 73.78 73.60 74.34 73.50 73.88 -0.519 -0.736
37891.25    73.78 73.60 74.34 73.50 73.89 -0.562 -0.693
37891.25347 73.78 73.65 74.34 73.50 73.89 -0.548 -0.693
37891.25694 73.78 73.65 74.34 73.50 73.89 -0.548 -0.693
37891.26042 73.78 73.60 74.34 73.50 73.90 -0.562 -0.693
37891.26389 73.56 73.39 73.82 73.49 73.78 -0.230 -1.299
37891.26736 73.00 72.78 72.91 73.49 73.53 -0.101 -1.945
37891.27083 72.57 72.31 72.39 73.48 73.32 -0.143 -2.161
37891.27431 72.22 72.01 72.09 73.47 73.17 -0.158 -2.332
37891.27778 72.01 71.79 71.92 73.47 73.06 -0.230 -2.331
37891.28125 71.88 71.66 71.79 73.45 72.96 -0.229 -2.374
37891.28472 71.75 71.58 71.71 73.44 72.89 -0.244 -2.374
37891.28819 71.66 71.53 71.58 73.44 72.82 -0.244 -2.416
37891.29167 71.53 71.40 71.49 73.44 72.76 -0.272 -2.416
37891.29514 71.45 71.32 71.40 73.43 72.71 -0.272 -2.459
37891.29861 71.36 71.32 71.36 73.42 72.67 -0.272 -2.458
37891.30208 71.32 71.28 71.32 73.41 72.63 -0.272 -2.458
37891.30556 71.23 71.19 71.19 73.41 72.58 -0.244 -2.544
37891.30903 71.19 71.15 71.19 73.41 72.53 -0.229 -2.544
37891.3125  71.15 71.10 71.15 73.41 72.50 -0.272 -2.501
37891.31597 71.06 71.06 71.10 73.40 72.47 -0.287 -2.500
37891.31944 71.02 71.02 71.02 73.39 72.43 -0.258 -2.586
37891.32292 70.97 71.02 70.97 73.38 72.41 -0.287 -2.543
37891.32639 70.93 70.97 70.97 73.39 72.37 -0.272 -2.543
37891.32986 70.89 70.93 70.97 73.38 72.36 -0.301 -2.543
37891.33333 70.85 70.93 70.93 73.37 72.32 -0.287 -2.543
37891.33681 70.85 70.89 70.85 73.36 72.30 -0.287 -2.586
37891.34028 70.80 70.85 70.85 73.36 72.28 -0.272 -2.585
37891.34375 70.76 70.85 70.85 73.37 72.27 -0.287 -2.585
37891.34722 70.72 70.80 70.80 73.37 72.24 -0.244 -2.628
37891.35069 70.72 70.76 70.76 73.38 72.23 -0.315 -2.585
37891.35417 70.67 70.76 70.72 73.37 72.19 -0.258 -2.628
37891.35764 70.63 70.72 70.72 73.38 72.17 -0.287 -2.628
37891.36111 70.59 70.72 70.67 73.39 72.16 -0.315 -2.585
37891.36458 70.59 70.72 70.63 73.38 72.15 -0.287 -2.628
37891.36806 70.54 70.63 70.63 73.37 72.11 -0.244 -2.671
37891.37153 70.54 70.63 70.63 73.37 72.11 -0.287 -2.628
37891.375   70.50 70.63 70.63 73.37 72.10 -0.301 -2.628
37891.37847 70.50 70.63 70.54 73.38 72.08 -0.287 -2.671
37891.38194 70.50 70.59 70.54 73.37 72.05 -0.301 -2.627
37891.38542 70.46 70.59 70.54 73.37 72.04 -0.272 -2.627
37891.38889 70.42 70.59 70.54 73.36 72.03 -0.287 -2.627
37891.39236 70.42 70.54 70.50 73.36 72.00 -0.272 -2.670
37891.39583 70.37 70.54 70.46 73.36 71.99 -0.258 -2.670
37891.39931 70.37 70.50 70.46 73.37 71.98 -0.315 -2.627
37891.40278 70.37 70.50 70.46 73.37 71.97 -0.272 -2.670
37891.40625 70.33 70.50 70.42 73.37 71.96 -0.301 -2.670
37891.40972 70.33 70.46 70.42 73.37 71.94 -0.272 -2.670
37891.41319 70.29 70.46 70.42 73.37 71.93 -0.287 -2.670
37891.41667 70.29 70.46 70.33 73.39 71.91 -0.272 -2.713
37891.42014 70.29 70.37 70.37 73.38 71.90 -0.287 -2.713
37891.42361 70.24 70.37 70.37 73.39 71.89 -0.301 -2.670
37891.42708 70.24 70.37 70.33 73.39 71.88 -0.315 -2.713
37891.43056 70.24 70.37 70.33 73.39 71.87 -0.315 -2.670
37891.43403 70.24 70.37 70.33 73.39 71.85 -0.272 -2.670
37891.4375  70.20 70.37 70.29 73.39 71.84 -0.258 -2.713
37891.44097 70.16 70.33 70.29 73.39 71.84 -0.329 -2.626
37891.44444 70.20 70.37 70.29 73.39 71.83 -0.301 -2.670
```