Abstract—Event-triggered control systems are systems in which the control signal is recomputed when the plant’s output signal leaves a triggering-set. There has been recent interest in event-triggered control systems as a means of reducing the communication load in control systems. This paper re-examines a problem [1] whose solution characterizes triggering-sets that minimize a quadratic control cost over a finite horizon subject to a hard constraint on the number of times the feedback control is computed. Computational complexity confined prior solutions of this problem to scalar linear systems. This paper presents an approximate solution that is suitable for multi-dimensional linear systems. This approximate solution uses families of quadratic forms to bound the value functions generated in solving the problem. This approach has a computational complexity that is polynomial in state-space dimension and horizon length. This paper’s results may therefore provide a basis for developing practical methods for the event-triggered output control of multi-dimensional discrete-time linear systems.

I. INTRODUCTION

There has been recent interest in event-triggered control systems. Event-triggered controllers adapt the real-time system’s task period directly in response to the application’s performance [2]. Under event-triggering, the control task is only executed when the application’s error signal leaves a specified triggering-set. Ostensibly, this error provides a measure of how valuable the current sensor data is in maintaining the overall system’s closed-loop performance. Since the system state is always changing, this approach generates an aperiodic sequence of controller invocations. In general, the hope is that the average rate of this aperiodic task set will be much lower than the rate of a periodic task set with comparable performance levels.

There is experimental evidence to support the assertion that event-triggered feedback can maintain performance levels while reducing feedback information. Results from [3] consider a controlled scalar diffusion process where control updates are triggered when the absolute value of the system state exceeds a specified constant threshold. These results show that the event-triggered system has better performance (in the sense of a lower steady state state variance) than a comparable system with periodically triggered control updates. Such results have helped stimulate interest in using event-triggering as a means of minimizing the feedback information used in achieving control objectives.

Recent work in [4], [5] has quantified the feedback rate in state-dependent event-triggered systems. The feedback rate is quantified by the intersampling interval; the time between consecutive samples. These papers analytically determine the minimum intersampling interval for event-triggered systems enforcing a specified stability concept such as input-to-state stability [4] or $L_2$ stability [5]. The determination of the associated intersampling interval, however, is done as an afterthought. In other words, these results first enforce the desired stability concept and then determine what the resulting intersampling interval will be. For some systems, this approach leads to a more efficient use of the feedback channel. It is also relatively easy, however, to use these methods to obtain event-triggered systems that exhibit Zeno-sampling [6]. In such cases the system generates an infinite sequence of intersampling intervals that asymptotically go to zero over a finite time interval. This leads to infinitely fast sampling of the system state, which clearly does not minimize the information rate in the feedback channel.

What is really needed is an analysis that treats control system performance and computational resource utilization within the same analytical framework. One well used framework treats the design of event-triggers as a constrained optimization problem that optimizes control system performance subject to a constraint on the feedback rate. This framework was used in [7] where the mean square control cost was minimized over an infinite horizon subject to a feedback rate constraint. Related finite horizon versions of this problem were considered in [1] and [8]. In general, these problems were all solved using dynamic programming methodologies that optimize system performance with respect to the event triggering-sets.

In practice, however, this framework was of limited utility for it was quickly recognized that the complexity of computing the optimal triggering-sets was impractical for multi-dimensional systems. As a result nearly all of the recent works [9], [10], [11] based on this framework have confined their attention to scalar linear systems. This restriction to scalar systems is of limited use in developing real-life applications of event-triggered systems. A major challenge to be addressed by the research community therefore lies in finding practical ways of extending this analytical framework to multi-dimensional systems.

First efforts at such multi-dimensional extensions were suggested in [12], [13] with respect to the infinite horizon problem posed in [7]. The approach in [12] used a single quadratic form to approximate the value function used in determining the optimal triggering-sets. This quadratic approximation was well suited to the infinite horizon problems in [7], but it was less effective in approximating the value
functions for the finite horizon problem in [1]. This is because the value functions for the finite-horizon problem may not be convex. Recent work [14] suggested that this problem could be addressed by using families of quadratic forms. The results presented below show how the suggestion in [14] might be used for event-triggered output feedback control over a finite horizon.

II. PROBLEM STATEMENT

Consider a linear discrete-time process over a finite horizon of length \( M + 1 \), during which only \( b \in \{0, 1, \ldots, M + 1\} \) transmissions are allowed. A block diagram of the closed-loop system is shown in figure 1. This closed-loop system consists of a discrete-time linear plant which generates measurement sequence, a sensor subsystem which processes the measurement sequence and decide when to transmit the processed data and an actuator subsystem which uses the information sent by sensor subsystem to compute the control signal.

The plant satisfies the difference equation below

\[
x_{k+1} = Ax_k + Bu_k + w_k, \\
y_k = Cx_k + v_k
\]

for \( k \in [0, 1, \ldots, M] \) where \( A \) is a \( n \times n \) real matrix, \( B \) is a \( n \times m \) real matrix, \( u \) is the control input, and \( w : [0, 1, \ldots, M] \to \mathbb{R}^n \) is a zero mean white noise process with covariance matrix \( Q \). The initial state, \( x_0 \), is a Gaussian random variable with mean \( \mu_0 \) and variance \( \Pi_0 \). \( y_k \) is the sensor measurement at time \( k \). \( C \) is a real \( p \times n \) matrix and \( v : [0, 1, \ldots, M] \to \mathbb{R}^n \) is another zero mean white noise process with variance \( R \). \( w, v \) and \( x_0 \) are uncorrelated with each other. We assume that \( (A, B, C) \) is controllable and observable. The sensor outputs are fed into a sensor subsystem that decides when to transmit information to the actuator subsystem.

The sensor-subsystem consists of three components: a filter, a local observer, and an event detector. Let \( Y_k = \{y_0, y_1, \ldots, y_k\} \) denote the sensor information available at time \( k \). The filter generates a state estimate \( \hat{x}_k \) that minimizes the mean square estimation error(MSSE) \( E[(x_k - \hat{x}_k)^2 \mid Y_k] \) at each time step conditioned on all of the sensor information received up to and including time \( k \). These estimates are computed using a Kalman filter. The filter equations for the system are,

\[
\begin{align*}
\hat{x}_k &= E[x_k \mid Y_k] = \hat{x}_k + L_k(y_k - C\hat{x}_k) \\
\hat{P}_k &= E[(x_k - \hat{x}_k)^2 \mid Y_k] = A\hat{P}_{k-1}A^T + Q - L_kC(A\hat{P}_{k-1}A^T + Q)
\end{align*}
\]

for \( k = 1, 2, \ldots, M \) where \( L_k \) is the Kalman filter gain and \( \hat{P}_k = A\hat{P}_{k-1} + Bu_{k-1} \). The initial condition \( \hat{P}_0 \) is the first a posteriori update based on \( y_0 \) and \( \hat{P}_0 \) is the covariance of this initial estimate.

Because the sensor subsystem has access to the information received by actuator subsystem, the local observer can duplicate the state estimate, \( \hat{x} \), made by the remote observer in the actuator subsystem. The behavior of local and remote observers will be explained later.

The event detector observes the filtered state, \( \hat{x}_k \) and the gap between filtered state and the remote estimated state, \( \hat{x}_k = \hat{x}_k - \hat{x}_k \). If the vector \( \begin{bmatrix} \hat{x}_k \\ \epsilon_k \end{bmatrix} \) lies outside the specified triggering set \( S_b^\epsilon \), where \( b \) is the remaining transmission times, the filtered state \( \hat{P}_k \) will be transmitted to the actuator subsystem. Given a set of transmission times \( \mathcal{T} = \{\tau_1, \tau_2, \ldots, \tau_{\mathcal{L}}(k)\} \) denote the filter estimates that were transmitted to the remote observer by time \( k \) where \( \ell(k) = \max\{\ell : \tau_\ell \leq k\} \). This is the information set available to the remote observer at time \( k \).

The actuator subsystem consists of two components: a remote observer and the controller gain. The remote observer uses the received information to compute an a posteriori estimate \( \hat{x} \) of the process state that minimizes the MSE, \( E[(x_k - \hat{x}_k)^2 \mid \hat{P}_k] \), at time \( k \) conditioned on the information received up to and including time \( k \). The a priori estimate of the remote observer, \( \hat{x}^- : [0, 1, \ldots, M] \to \mathbb{R}^n \), minimizes \( E[(x_k - \hat{x}_k)^2 \mid \hat{P}_{k-1}] \), the MSE at time \( k \) conditioned on the information received up to and including time \( k - 1 \). These estimates take the form

\[
\begin{align*}
\dot{x}_k^- &= E[x_k \mid \hat{P}_{k-1}] = A\hat{x}_{k-1} + Bu_{k-1} \\
\hat{x}_k^- &= E[x_k \mid \hat{P}_k] = \{ \begin{array}{ll} \hat{x}_k^- & \text{don’t transmit at step } k \\
\hat{x}_k & \text{transmit at step } k \end{array} \}
\end{align*}
\]

where \( \hat{x}_k^- = \mu_0 \). This estimate is then used to compute the control, \( u_k = K\hat{x}_k \), for \( k = 0, 1, \ldots, M \) where \( K \) is some real \( m \times n \) matrix.

For convenience, we let

\[
S_b^\epsilon(k) = \{ S_{k}^{\epsilon_{max}(0,b-k+r)}, \ldots, S_{k}^{\epsilon_{min}(b,M+1-k)} \}
\]

denote the triggering sets to be used at step \( k \) with \( b \) transmissions remaining. We let \( S_b^\epsilon = \{ S_b^\epsilon(1), \ldots, S_b^\epsilon(M) \} \) be the collection of all triggering-sets that will be used by the sensor-subsystem after and including time step \( r \).

We are now in a position to formally state the problem being addressed. Consider the cost function

\[
J_M(\hat{P}_0) = E \left[ \sum_{k=0}^{M} p_k^T Z p_k \right]
\]
where $Z = \begin{bmatrix} Z_{11} & Z_{12} \\ Z_{21} & Z_{22} \end{bmatrix}$ is a symmetric and positive semi-definite $2n$ by $2n$ matrix and $p_k = \begin{bmatrix} x_k \\ \hat{e}_k \end{bmatrix}$ is the system state at time $k$, where $\hat{e}_k = x_k - \hat{x}_k$, is the remote state estimation error. The objective is to find the collection, $S^N_0$, of triggering-sets that minimizes the cost function. The optimal cost then becomes

$$J^*_M = \min_{S^N_0} J_M(S^N_0)$$

III. MAIN RESULTS

The problem is an optimal control problem whose controls are the triggering-sets in $S^N_0$. The solution may be characterized using dynamic programming techniques. Define the problem’s value function as

$$V(\theta, b; r) = \min_{S^r_0} \left( \sum_{k=r}^{M} p_k^T Z p_k | I_r = (\theta, b) \right).$$

For convenience, indicate $\left[ \begin{array}{c} \pi_r \\ e_r \end{array} \right]$ by $q_r$ and $\left[ \begin{array}{c} \pi_r \\ e_r \end{array} \right]$ by $q_r$. $I_r$ is the a priori information set at time step $r$ consisting of an ordered pair $(q_r, b)$ with $b$ remaining transmissions. The value function is defined as the minimum cost conditioned on $q_r = \theta = \begin{bmatrix} \eta \\ \zeta \end{bmatrix}$ with $b$ remaining transmissions.

Theorem 3.1: The value function satisfies

$$V(\theta, b; r) = \min \left\{ V_{nt}(\theta, b, r), V_t(\theta, b, r) \right\}$$

where $V_{nt}$ is the cost function without transmitting at step $r$ and $V_t$ is the cost function if transmitting at step $r$. Both of them are defined as

$$V_{nt}(\theta, b, r) = E \left[ V(q_{r+1}, b; r+1) | I_r = (\theta, b) \right] + \theta^T Z \theta + \beta_r$$

and

$$V_t(\theta, b, r) = E \left[ V(q_{r-1}, b-1; r+1) | I_r = (\theta_0, b-1) \right] + \theta_0^T Z \theta_0 + \beta_r$$

where $I_r$ is the posteriori information set with ordered pair $(q_r, b)$. $\theta = \begin{bmatrix} \eta \\ \zeta \end{bmatrix}$ and $\theta_0 = \begin{bmatrix} \eta \\ 0 \end{bmatrix}$ are the actual values of a posteriori random variable, $q_r$. The scalar $\beta_r$ equals $\text{tr}(P_r (Z_{11} + Z_{12} + Z_{21} + Z_{22}))$.

This theorem indicates that the value function choose the smaller one between these two cost functions.

The preceding theorem shows that $V(\theta, b, r)$ can be computed through a recursion that ranges over the indices $b$ (number of remaining transmissions) and $r$ (current time). The initial conditions for this recursion occur when $b = 0$ or $b = M + 1 - r$ for all values of $r$. For the first case ($b = 0$), this corresponds to the cost of never transmitting after time step $r$. The other case ($b = M + 1 - r$) corresponds to transmitting at every single remaining time step. In both cases, the value function can be computed in closed form, and the expressions are given in appendix.

Given these initial conditions, the value function at index $(b, r)$ may be computed from the value function at indices $(b, r + 1)$ and $(b - 1, r + 1)$. This computational dependence in the recursion is illustrated in figure 2. This figure shows the indices including the triggering set collection $S^N_1$. The indices for the initial value functions are filled in. The order of computation used to compute $S^N_1$ is shown by the arrows.

The selection in equation (1) defines the triggering-sets used by the event detector.

**Corollary 3.2:** The optimal triggering-set used at time step $r$ with $b$ transmissions remaining will be

$$S^b_r(\eta) = \{ \theta = \begin{bmatrix} \eta \\ \zeta \end{bmatrix} | V_{nt}(\theta, b, r) \leq V_t(\theta, b, r) \}$$

The initial triggering-sets are $S^{0\theta}_r = \mathbb{R}^{2n}$ and $S^{(M+1-r)\theta}_r = \emptyset$.

The recursion used in equations (2) and (3) may only be tractable for first order linear systems. In this case, the triggering sets are subsets of $\mathbb{R}^2$ and the bisection search from [14] may be employed to determine the triggering-sets $S^\theta_r$. This is done for a specific example below. Extending this approach to multi-dimensional systems is impractical. The approach used in [14] involves computing the value function over a grid of points in the state space. Overall, there are $\mathcal{O}(M + 1 - b)$ triggering sets in the collection $S^N_0$. If each value function is evaluated in a $2n$-dimensional space over a range of $[-c/2, c/2]$ with a granularity of $\epsilon$, then there are a total of $(\frac{c}{\epsilon})^{2n}$ points at which the value functions are computed. This means the computational effort required to compute $V(\theta, b; r)$ will be on the order of $\mathcal{O} \left( \frac{c}{\epsilon} (M + 1 - b) \right)$.

The initial triggering-sets are $S^{0\theta}_r = \mathbb{R}^{2n}$ and $S^{(M+1-r)\theta}_r = \emptyset$.

Since the computational complexity of the recursion in equations (2) and (3) will be prohibitively large, one must resort to approximation methods. One useful approximation [12] was developed for the infinite horizon problem considered in [7]. This approximation used a single quadratic form to over bound the value function. While this method works well for infinite horizon problems, it seems to be ill-suited for finite horizon problems. In particular, recent work [14] for the finite horizon estimation problem [15] shows that the value functions are non-convex and are therefore...
poorly approximated by a single quadratic form. The work in [14] suggested that a family of quadratic forms provide a much better way of approximating the value function for the estimation problem. This approach can also be adopted for the output feedback control problem considered in this paper.

The basic idea behind the approximations used in [14] is as follows. While the value function, , is inherently non-convex due to the choice in equation 1, the functions may be well approximated by quadratic forms. This conjecture is based on two observations. First the initial value functions for and are quadratic and second that the recursion in equations (2) and (3) is nearly quadratic. It therefore seems possible that we can bound from above by a family of quadratic forms.

Proposition 3.3: There exist , , and scalars , for , and , and such that

where

\[ V_{nt}(\theta, b, r) \leq \nabla_{nt}(\theta, b, r) = \min_{j \in [1, \ldots, p]} \{ \theta^T \Lambda_{r,j} \theta + c_{r,j} \} \]  

(5)

\[ V_t(\theta, b, r) \leq \nabla_t(\theta, b, r) = \eta^T \Psi b_t \eta + d_{r,t} \]  

(6)

where \( \rho^b_r \) is a finite integer associated with step \( r \) and remaining transmissions \( b \).

With the upper bounds of the true value functions, \( \nabla_{nt} \) and \( \nabla_t \), we can construct a sub-optimal triggering-set \( S^{b^+}_r \) of the form

\[ S^{b^+}_r = \{ \theta \in \mathbb{R}^{2n} : \nabla_{nt}(\theta, b, r) \leq \nabla_t(\theta, b, r) \} \]  

(7)

which is an approximation of the optimal triggering-sets, \( S^{b^+}_r \), in equation (4).

We notice that (2) and (3) add a quadratic value to the expected minimum of \( V_t \) and \( V_{nt} \). The approximation can be done by interchanging the expectation and minimization operators as \( V_{nt} = \theta^T Z \theta + \beta + E[\min(V_t, V_{nt})] \leq \theta^T Z \theta + \beta + \min \{ E[V_t], E[V_{nt}] \} \), where the expected values can again be represented by a family of quadratic forms. Provided the variances of the noise processes are relatively small, this approximation can be made tight.

For convenience, we let \( \mathbf{A} = \begin{bmatrix} A + BK & -BK \\ 0 & A \end{bmatrix} \),

\[ \mathbf{T}_k = \begin{bmatrix} L_k & L_k \\ L_k & L_k \end{bmatrix}, \quad \beta_k = \text{tr}(P_k(Z_{11} + Z_{12} + Z_{21} + Z_{22})) \]

and \( S_k = C \mathbf{A}^T \mathbf{T}_k C^T + CQC^T + R \). It can be easily shown by using mathematical induction and the fact that \( E[\min(V_t, V_{nt})] \leq \min \{ E[V_t], E[V_{nt}] \} \) that

Lemma 3.4: Equation (5) and (6) hold, if for all \( b \geq 1 \) and all \( b - b \leq r \leq M - b \),

\[ \Lambda^b_{r-j} = \begin{cases} Z + \mathbf{A}^T \Lambda^b_{r-j+1} \mathbf{A} & j = 1, \ldots, \rho^b_r \\ Z + \mathbf{A}^T \Psi^b_{r+1} \mathbf{A} & j = 0 \end{cases} \]  

(8)

In this case, \( \rho^b_r \) equals \( M + 1 - b - r \) for \( b \geq 1 \), and 1 for \( b = 0 \). The initial condition is the same as defined in theorem 3.1.

Because the recursion used above mimics the recursions used for the original value function, we expect these bounds to be relatively tight. Precisely how tight these bounds are is still being quantified.

Computing the suboptimal triggering-sets involves a \( 2n \) by \( 2n \) matrix-matrix multiplication with a computational complexity \( O((2n)^3) \). The computation of \( V_{nt} \) dominates the effort since it has the most quadratic forms to compute. One can therefore show that the effort associated with computing the suboptimal triggering set \( S^{b^+}_r \) will be \( O(\tilde{b}(M + 1 - \tilde{b})(M + 2 - \tilde{b})(2n)^3) \). This has a complexity that is polynomial in \( n \) and quadratic in \( M \) (the length of the horizon window). The complexity is much lower than that used in computing the value functions, so these approximations may represent a practical way of implementing optimal event-triggered controllers provided the approximations are tight. Preliminary simulation results are given below to experimentally evaluate how good the approximation really is.

IV. PRELIMINARY SIMULATION RESULTS

As stated above, we’d like to experimentally evaluate how closely the approximations in equations (5) and (6) approximate the value function computed using the equations (2) and (3). We’ll do this for a specific example. Because we can only compute the exact value function for scalar systems, this example focuses only on the scalar system.

The system under study is a scalar system where \( A, B, C, D = 1, Q, R = 1, \mu_0, \mu_1 = 1, K = -9.5, M = 4 \) and \( \tilde{b} = 1 \). We consider a control problem without a penalty on the control input, so that \( Z = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \). The value functions and their bounds were computed using the recursions described in the preceding section. The results from this comparison are shown below in figure 3.

The left column of the top plots in figure 3 shows the value functions and their upper bounds. While it may be difficult to see, both the value function and the upper bound are shown in these graphs. If one looks closely along the plane where \( \eta = 0 \), one may see a white line that marks the upper bound. For \( k = 0 \) and \( k = 1 \), these plots show a small
Fig. 3. Top plots show value functions and optimal/sub-optimal triggering sets. Bottom plot shows experiment results.
25 \times 10^5$ whereas the space-complexity of the suboptimal approach is $10(2n)^2 = 40$. The cost of evaluating an event-trigger for the optimal case is 400 whereas the suboptimal case only requires 20 multiplies. For this example, the proposed suboptimal method clearly has a much smaller computational cost than the optimal method. Moreover, the suboptimal thresholds work nearly as well as the optimal ones as indicated in the bottom plot of figure 3.

V. SUMMARY

This paper presents a computationally tractable approach for determining suboptimal event-triggers in finite-horizon output-feedback problems. The approach relies on using a family of quadratic forms to characterize the value functions in the problem's optimal dynamic program. Our example shows that this sub-optimal sets is much more computational effective and have the similar performance as the optimal triggering sets.

VI. APPENDIX

**Lemma 6.1:** \( \{I_k^-, I_k\}_{k=0}^M \) is Markov.

**Proof:** 
\[
I_k = \left\{ \begin{array}{ll}
\eta_k I_k^-, & 0 \leq \eta_k \leq s_k^h,

\bar{\eta}_k - 1, & \eta_k > s_k^h,
\end{array} \right. 
\]

So it's easy to see that \( p(I_k | I_k^-, \cdots, I_0^-, I_0) = p(I_k | I_0^-) \).

\( I_{k+1}^- = (\bar{A}_{k+1} \eta_{k+1} + \bar{T}_{k+1} \bar{\eta}_k, 0) \) \( \bar{\eta}_k = CA_{k+1} \bar{\eta}_k + C\bar{u}_{k+1} + \bar{u}_{k+1}, \bar{e}_k = x_k - \bar{X}_k \) is the local state estimation error. Because \( \bar{\eta}_k \) is independent with \( I_k, I_k^-, \cdots, I_0^- \). So \( p(I_{k+1}^- | I_k^-, \cdots, I_0^-) = p(I_{k+1}^-. | I_k) \).

**Proof of theorem 3.1**

**Proof:** By the definition of the value function, we have

\[
V(\theta, b, r) = \min_{S^b_r} (V_{nt}(\theta, b, r)1_{\theta \in S^b_r}, V_I(\theta, b, r)1_{\theta \in S^b_r}),
\]

where

\[
V_{nt}(\theta, b, r) = \min_{S^b_r} \mathbb{E} \left( \sum_{k=r}^{M} p_k^T Z p_k | I_r^- = (\theta, b) \right) 1_{\theta \in S^b_r},
\]

\[
V_I(\theta, b, r) = \min_{S^b_r} \mathbb{E} \left( \sum_{k=r}^{M} p_k^T Z p_k | I_r^- = (\theta, b) \right) 1_{\theta \in S^b_r}.
\]

Consider \( V_{nt} \) first

\[
V_{nt}(\theta, b, r) = \min_{S^b_r} \mathbb{E} \left( \sum_{k=r}^{M} p_k^T Z p_k | I_r^- = (\theta, b) \right) 1_{\theta \in S^b_r}.
\]

The condition is equivalent with \( I_r^- = I_r = (\theta, b) \), because \( \theta \in S^b_r \) means no transmission at step \( r \). With lemma 6.1, we can derive that

\[
V_{nt}(\theta, b, r) = \min_{S^b_r} \mathbb{E} \left( \sum_{k=r}^{M} p_k^T Z p_k | I_r^- = (\theta, b) \right) 1_{\theta \in S^b_r}.
\]

We can pull the cost at the \( r \)th step out of the minimum, and the remaining costs are only related with \( S^b_{r+1} \), so

\[
V_{nt}(\theta, b, r) = \theta^T Z \theta + \beta_r + \min_{S^b_{r+1}} \mathbb{E} \left( \sum_{k=r+1}^{M} p_k^T Z p_k | I_r^- = (\theta, b) \right)
\]

With lemma 6.1 and some mathematical deduction, we are able to show equation (2) holds.

Follow the same steps, (3) can be shown.

Initial conditions are given for two cases: \( b = 0 \) and \( b + r = M + 1 \). For the first case, \( V_{nt}(\theta, 0, r) = \theta^T \Lambda^0_r \theta + c^0_r \) and \( V_I(\theta, 0, r) = \eta^T \eta + d^0_r \) where \( \Lambda^0_r = \sum_{k=r}^{M} (A^T)^k - r Z ((A^T)^{k-r} - \bar{\Lambda})^{-1} Z ((A^T)^{k-r} - \bar{\Lambda})^{-1} \).

**REFERENCES**


