Journal of Theoretical and Applied Vibration and Acoustics 1(2) 96-106 (2015)

I S A V

Journal of Theoretical and Applied

Vibration and Acoustics

journal homepage: http://tava.isav.ir

Optimal integrated passive/active design of the suspension system using iteration on the Lyapunov equations

Mahyar Naraghi *, **Mojtaba Moradi**

Department of Mechanical Engineering, Amirkabir University of Technology, Tehran, Iran

KEYWORDS

Integrated active/passive design

Suspension system

Optimal control

Riccati equation

Lyapunov equation

ABSTRACT

In this paper, an iterative technique is proposed to solve linear integrated active/passive design problems. The optimality of active and passive parts leads to the nonlinear algebraic Riccati equation due to the active parameters and some associated additional Lyapunov equations due to the passive parameters. Rather than the solution of the nonlinear algebraic Riccati equation, it is proposed to consider an iterative solution method based on the Lyapunov equations in the Newton optimization scheme for both active and passive parameters. The main contribution of the paper is considered as the concept that it doesn't require to optimize controller when the plant is not optimal. The proposed method is verified by designing a one-quarter active suspension system. The results indicate that the algorithm is more efficient as compared to solving the problem through the direct Riccati solution based method while its derivation and application is simple. Significant improvements can be seen in comparison to the previous method.

1. Introduction

Considering the coupling between passive dynamics (the physical system) and active dynamics (the associated control system) plays a critical role in the modern design methodology and it is called Integrated Passive and Active Design (IPAD). The term 'integrated' means simultaneous consideration of methodologies in the design. The IPAD connects the passive and active designs whereas they were separated and conventionally designed in the sequential manner. In this sequential method, first, the passive parameters are designed. Then the passive variable set is considered fixed and the controller is designed [1]. It is prominent that in some cases, the IPAD enables solving some previously unsolvable problems especially for systems with strong coupling such as flexible robotics and vibration control systems.

Optimization is the main methodology of the engineering design and it can be considered as a main paradigm in the IPAD. From the mathematical point of view, it can be stated as a deterministic optimal control problem (OCP) with some additional subproblems related to the

* Corresponding Author: Mahyar Naraghi, Email: naraghi@aut.ac.ir

design of passive parameters. For full information, in an unconstrained, infinite horizon, linear system with a quadratic cost function, the optimal control becomes a linear feedback of the states associated with the standard Algebraic Riccati Equation (ARE). Based on the OCP, the IPAD problem was first considered by Hale et al. [2] for flexible satellite control in where an open-loop control scheme was applied. As an indirect and ARE-based OCP, Salama et al. [3] eliminated control design variables by considering sensitivity equations of ARE with respect to the parameters, namely Lyapunov equations (LE). This method is extended by Haftka et al. [4] to minimize the control effort through structural changes while maintaining specified damping ratios. Bodden and Junkins [5] minimized some unspecified robustness measures while placing closed-loop eigenvalues in a desired region. Canfield and Meirovitch [6] used mode decomposition for simplifying the structure to a number of two order systems and by using optimization problem for these systems through LE, still solved the OCP by the ARE. Alisson and Han [7] proposed simulation based IPAD that needs integration to compute the cost. From the analytical point of view, any simulation-based method should be avoided since they need several computations due to the integration.

In this paper, an extension to the well-known Kleinman's [8] method for the solution of ARE based on the Newton algorithm is proposed to solve the IPAD problem with application to the suspension system. The Kleinman's algorithm uses a simple LE solution to improve the controller. In the Salama's iteration [3], the solution of ARE and some sensitivities with respect to the design parameters are needed. Since in every step the plant is not optimal, there is no evidence to optimize the controller. This concept has motivated the authors to use sensitivity of LE associated with the Kleinman's algorithm in control improvement. The proposed method presents a very simpler equation in comparison to Salama iterations. The method is intentionally simple and easy to apply and its efficiency is shown by simulation.

2. Problem Formulation

2.1. Dynamic equations of the linear mechanical system

Consider a linear mechanical system as a plant subjected to the known initial conditions $\mathbf{q}(0)=\mathbf{q}_0$ and $\dot{\mathbf{q}}(0)=\dot{\mathbf{q}}_0$. Dynamic equations which include a force vector $\mathbf{u}(t)\in\mathbb{R}^{m}$ for controlling the dynamic response $\mathbf{q}(t)\in\mathbb{R}^{nq}$ are:

$$\mathbf{M}(\mathbf{p})\ddot{\mathbf{q}}+\mathbf{C}(\mathbf{p})\dot{\mathbf{q}}+\mathbf{K}(\mathbf{p})\mathbf{q}=\mathbf{E}(\mathbf{p})\mathbf{u} \tag{1}$$

where $\mathbf{M}$, $\mathbf{C}$ and $\mathbf{K}\in\mathbb{R}^{nq\times nq}$ denote the mass, damping and stiffness matrices respectively, $\mathbf{E}\in\mathbb{R}^{nq\times m}$ denotes the actuator distribution matrix and $\mathbf{p}\in\mathbb{R}^{np}$ denotes the parameters vector. Eq. (1) can be rewritten in the state space form as,

$$\dot{\mathbf{x}}=\mathbf{A}(\mathbf{p})\mathbf{x}+\mathbf{B}(\mathbf{p})\mathbf{u} \tag{2}$$

where $\mathbf{x}\in\mathbb{R}^{n}$ ($n=2n_q$) denotes the state vector defined by $\mathbf{x}=(\mathbf{q}^T, \dot{\mathbf{q}}^T)^T$ while $\mathbf{x}_0=(\mathbf{q}_0^T, \dot{\mathbf{q}}_0^T)^T$ is the known initial state, $\mathbf{A}\in\mathbb{R}^{n\times n}$ and $\mathbf{B}\in\mathbb{R}^{n\times m}$ are the system and the control matrices stated as follows.

$$\mathbf{A}(\mathbf{p})=\left[\begin{array}{c|c}\mathbf{0} & \mathbf{I}\\ \hline -\mathbf{M}^{-1}\mathbf{K} & -\mathbf{M}^{-1}\mathbf{D}\end{array}\right],\mathbf{B}(\mathbf{p})=\left[\begin{array}{c}\mathbf{0}\\ \hline \mathbf{M}^{-1}\mathbf{E}\end{array}\right] \tag{3}$$

Classic design of systems are performed by optimization of the plant first, and then, the optimal control for the optimized plant is considered which can be stated as,

$$V^{*}_{\text{Classic}} = \min_{\mathbf{p}} J_{\text{P}}(\mathbf{p}) + \min_{\mathbf{u}} J_{\text{C}}(\mathbf{p}^{*}, \mathbf{u}) \tag{4}$$

where V denotes the overall cost, J denotes the cost function with subscript P for the passive parameters and C for the controller.

Performing the IPAD means combining the above separated optimizations as,

$$V^{**}_{\text{IPAD}} = \min_{\mathbf{p},\mathbf{u}} \left[J_{\text{P}}(\mathbf{p}) + J_{\text{C}}(\mathbf{p}, \mathbf{u}) \right] \tag{5}$$

The IPAD for linear systems is reduced to finding a static feedback controller that minimizes the next quadratic performance index in the control and the state as,

$$V(\mathbf{x}_0) = w_1 J_{\text{P}}(\mathbf{p}) + w_2 \int_0^{\infty} \left(\mathbf{x}^T \mathbf{Q}(\mathbf{p}) \mathbf{x} + \mathbf{u}^T \mathbf{R}(\mathbf{p}) \mathbf{u} \right) dt \tag{6}$$

subject to Eq. (2) where $\mathbf{Q} \in \mathbb{R}^{n\times n}$ and $\mathbf{R} \in \mathbb{R}^{m\times m}$ are positive definite weighing matrices, $w_{1,2} \geq 0$ are compromising parameters that should be selected properly to define a truthful cost.

2.2. Cost evaluation

The controller is considered as a linear feedback as $\mathbf{u} = -\mathbf{k}\mathbf{x}$ and the fitness (6) can be restated as

$$V_{\mathbf{k}_0}(t) = m(\mathbf{p}) + \int_0^{\infty} \mathbf{x}^T \left(\mathbf{Q} + \mathbf{k}^T \mathbf{R} \mathbf{k} \right) \mathbf{x}\, dt \tag{7}$$

where $m(\mathbf{p}) = \frac{w_1}{w_2} J_{\text{P}}(\mathbf{p})$. Substitution of the closed loop trajectory $\mathbf{x} = \mathbf{x}_0 e^{(\mathbf{A} - \mathbf{B}\mathbf{k})t}$ where $\mathbf{x}_0$ is the test initial state in the Eq. (7), leads to,

$$V_{\mathbf{k}_0}(t) = m(\mathbf{p}) + \mathbf{x}_0^T \mathbf{P} \mathbf{x}_0 \tag{8}$$

where the matrix $\mathbf{P} \geq 0$ is satisfying the following LE,

$$\mathbf{0} = (\mathbf{A} - \mathbf{B}\mathbf{k})^T \mathbf{P} + \mathbf{P}(\mathbf{A} - \mathbf{B}\mathbf{k}) + \mathbf{Q} + \mathbf{k}^T \mathbf{R} \mathbf{k} \tag{9}$$

where it can be solved by using efficient numerical methods (see the introduction). It is well-known that the application of the necessary conditions w.r.t the gain matrix $\mathbf{k}$ expresses its optimum as the Kalman's gain,

$$\mathbf{k}_{\infty} = -\mathbf{R}^{-1} \mathbf{B}^T \mathbf{P}_{\infty} \tag{10}$$

where $\mathbf{P}_{\infty}$ is the solution of the following ARE,

$$\mathbf{A}^T \mathbf{P}_{\infty} + \mathbf{P}_{\infty} \mathbf{A} - \mathbf{P}_{\infty} \mathbf{B}^T \mathbf{R}^{-1} \mathbf{B} \mathbf{P}_{\infty} + \mathbf{Q} = 0 \tag{11}$$

2.3. Salama iteration to find of optimal parameters

Local minimization of (8) w.r.t the parameter vector $\mathbf{p}$ can also be applied by using Salama [3] iteration as,

$$\mathbf{p}^{i+1} = \mathbf{p}^{i} - \alpha \left[\nabla^2 \left(m + \mathbf{x}_0^T \mathbf{P} \mathbf{x}_0 \right) \right]^{-1} \nabla \left(m + \mathbf{x}_0^T \mathbf{P} \mathbf{x}_0 \right) \tag{12}$$

where ∇ is the gradient operator, α is the convergence rate adjusting parameter and it should be lower than a certain value [3].

The gradient and hessian matrices can be stated as,

$$\begin{aligned}\nabla\left(m+\mathbf{x}_0^T\mathbf{P}\mathbf{x}_0\right)&=\nabla m+\nabla\left(\mathbf{x}_0^T\mathbf{P}\mathbf{x}_0\right)=\left[\frac{\partial m}{\partial p_i}+\mathbf{x}_0^T\frac{\partial \mathbf{P}}{\partial p_i}\mathbf{x}_0\right],\\ \nabla^2\left(m+\mathbf{x}_0^T\mathbf{P}\mathbf{x}_0\right)&=\nabla^2 m+\nabla^2\left(\mathbf{x}_0^T\mathbf{P}\mathbf{x}_0\right)=\left[\frac{\partial^2 m}{\partial p_j\partial p_i}+\mathbf{x}_0^T\frac{\partial^2 \mathbf{P}}{\partial p_j\partial p_i}\mathbf{x}_0\right],\end{aligned}\tag{13}$$

where the term $\frac{\partial \mathbf{P}}{\partial p_i}$ can be found by taking derivation of ARE(11) as follows,

$$\mathbf{A}_c^T\frac{\partial \mathbf{P}}{\partial p_i}+\frac{\partial \mathbf{P}}{\partial p_i}\mathbf{A}_c+\mathbf{Q}'=0\tag{14}$$

where

$$\begin{aligned}\mathbf{A}_c&=\mathbf{A}+\mathbf{B}\mathbf{k}\\ \mathbf{Q}'&=\mathbf{A}_{c,i}^T\mathbf{P}+\mathbf{P}\mathbf{A}_{c,i}+\frac{\partial \mathbf{Q}}{\partial p_i}\\ \mathbf{A}_{c,i}&=\frac{\partial \mathbf{A}}{\partial p_i}+\frac{\partial \mathbf{B}}{\partial p_i}\mathbf{k}-\mathbf{B}\mathbf{R}^{-1}\frac{\partial \mathbf{R}}{\partial p_i}\mathbf{k}-\mathbf{B}\mathbf{R}^{-1}\frac{\partial \mathbf{B}^T}{\partial p_i}\mathbf{P}\end{aligned}\tag{15}$$

It is evident that the second derivation can be found by repeating the above process. It follows,

$$\mathbf{A}_c^T\frac{\partial^2 \mathbf{P}}{\partial p_j\partial p_i}+\frac{\partial^2 \mathbf{P}}{\partial p_j\partial p_i}\mathbf{A}_c+\mathbf{Q}''=0\tag{16}$$

where

$$\begin{aligned}\mathbf{Q}''&=\frac{\partial^2 \mathbf{A}^T}{\partial p_j\partial p_i}\mathbf{P}+\mathbf{A}_{c,j}^T\frac{\partial \mathbf{P}}{\partial p_i}+\frac{\partial \mathbf{P}}{\partial p_i}\mathbf{A}_{c,j}+\mathbf{A}_{c,i}^T\frac{\partial \mathbf{P}}{\partial p_j}+\frac{\partial \mathbf{P}}{\partial p_j}\mathbf{A}_{c,i}+\mathbf{P}\mathbf{A}_{c,ji}+\frac{\partial^2 \mathbf{Q}}{\partial p_j\partial p_i}\\ \mathbf{A}_{c,j}&=\frac{\partial \mathbf{A}}{\partial p_j}+\frac{\partial \mathbf{B}}{\partial p_j}\mathbf{k}-\mathbf{B}\mathbf{R}^{-1}\frac{\partial \mathbf{R}}{\partial p_j}\mathbf{k}-\mathbf{B}\mathbf{R}^{-1}\frac{\partial \mathbf{B}^T}{\partial p_j}\mathbf{P},\\ \mathbf{A}_{c,ji}&=\frac{\partial^2 \mathbf{A}}{\partial p_j\partial p_i}+\mathbf{B}\mathbf{R}^{-1}\frac{\partial \mathbf{R}}{\partial p_j}\mathbf{R}^{-1}\frac{\partial \mathbf{B}^T}{\partial p_i}\mathbf{P}-\mathbf{B}\mathbf{R}^{-1}\frac{\partial^2 \mathbf{B}^T}{\partial p_j\partial p_i}\mathbf{P}-\frac{\partial \mathbf{B}}{\partial p_j}\mathbf{R}^{-1}\frac{\partial \mathbf{B}^T}{\partial p_i}\mathbf{P}\\ &+\mathbf{B}\mathbf{R}^{-1}\frac{\partial \mathbf{R}}{\partial p_i}\mathbf{R}^{-1}\frac{\partial \mathbf{B}^T}{\partial p_j}\mathbf{P}+\mathbf{B}\mathbf{R}^{-1}\frac{\partial \mathbf{R}}{\partial p_i}\mathbf{R}^{-1}\frac{\partial \mathbf{R}}{\partial p_j}\mathbf{k}-\mathbf{B}\mathbf{R}^{-1}\frac{\partial^2 \mathbf{R}}{\partial p_j\partial p_i}\mathbf{k}-\frac{\partial \mathbf{B}}{\partial p_i}\mathbf{R}^{-1}\frac{\partial \mathbf{R}}{\partial p_j}\mathbf{k}\\ &-\mathbf{B}\mathbf{R}^{-1}\frac{\partial \mathbf{R}}{\partial p_j}\mathbf{R}^{-1}\frac{\partial \mathbf{R}}{\partial p_i}\mathbf{k}-\frac{\partial \mathbf{B}}{\partial p_j}\mathbf{R}^{-1}\frac{\partial \mathbf{R}}{\partial p_i}\mathbf{k}-\frac{\partial \mathbf{B}}{\partial p_i}\mathbf{R}^{-1}\frac{\partial \mathbf{B}^T}{\partial p_j}\mathbf{P}+\frac{\partial^2 \mathbf{B}}{\partial p_j\partial p_i}\mathbf{k}.\end{aligned}\tag{17}$$

Not surprisingly, Eqs. (14) and (16) become the LEs.

2.4. The proposed method: Extended Kleinman iteration

Since the ARE is a nonlinear matrix equation in **P**, it is mostly preferred to use the Kleinman iteration by iteratively solving of LE (9) which is linear in **P** and then improving the controller by Eq. (10). The restriction to this method is the necessity of the stability insurance in the initialization step. In IPAD, in each iteration of Eq. (12), the passive parameters are altered and thus the stability check should be done in each step (as in applied using barrier function in the section 2.4.1). The Kleinman algorithm inherently is the Newton iteration [9]. Therefore by mixing of the two Newton iteration one for the passive parameters optimization and next for the controller, the IPAD can be applied. The major advantage of this solution is getting involved with linear system of equations comparing with (11) the simpler equations comparing with (14) and (16) which leads to have a smoother variations. Derivation of LE(9) w.r.t the parameter p_{i} leads to the following LEs:

$$0=\left(\mathbf{A}-\mathbf{B}\mathbf{k}\right)^{T}\frac{\partial\mathbf{P}}{\partial p_{i}}+\frac{\partial\mathbf{P}}{\partial p_{i}}\left(\mathbf{A}-\mathbf{B}\mathbf{k}\right)+\mathbf{Q}' \tag{18}$$

where $\mathbf{Q}'=\mathbf{A}_{c,i}^{T}\mathbf{P}+\mathbf{P}\mathbf{A}_{c,i}+\frac{\partial\mathbf{Q}}{\partial p_{i}}+\mathbf{k}^{T}\frac{\partial\mathbf{R}}{\partial p_{i}}\mathbf{k}$ and $\mathbf{A}_{c,i}=\frac{\partial\mathbf{A}}{\partial p_{i}}-\frac{\partial\mathbf{B}}{\partial p_{i}}\mathbf{k}$. Note that Eq. (18) is less complicated than Eq (14). To find the hessian matrix, another derivation is taken from Eq. (18) w.r.t the parameter p_{j} and then,

$$0=\left(\mathbf{A}-\mathbf{B}\mathbf{k}\right)^{T}\frac{\partial^{2}\mathbf{P}}{\partial p_{j}\partial p_{i}}+\frac{\partial\mathbf{P}}{\partial p_{j}\partial p_{i}}\left(\mathbf{A}-\mathbf{B}\mathbf{k}\right)+\mathbf{Q}''_{ij} \tag{19}$$

where $\mathbf{Q}''_{ij}=\mathbf{A}_{c,\mathrm{j}}^{T}\frac{\partial\mathbf{P}}{\partial p_{i}}+\frac{\partial\mathbf{P}}{\partial p_{i}}\mathbf{A}_{c,j}+\mathbf{A}_{c,i}^{T}\frac{\partial\mathbf{P}}{\partial p_{j}}+\frac{\partial\mathbf{P}}{\partial p_{j}}\mathbf{A}_{c,i}+\mathbf{A}_{c,ji}^{T}\mathbf{P}+\mathbf{P}\mathbf{A}_{c,ji}+\frac{\partial^{2}\mathbf{Q}}{\partial p_{j}\partial p_{i}}+\mathbf{k}^{T}\frac{\partial^{2}\mathbf{R}}{\partial p_{j}\partial p_{i}}\mathbf{k}$ and $\mathbf{A}_{c,ji}=\frac{\partial^{2}\mathbf{A}}{\partial p_{j}\partial p_{i}}-\frac{\partial^{2}\mathbf{B}}{\partial p_{j}\partial p_{i}}\mathbf{k}$.

This application of necessary condition needs to solve number of n_p LEs for first and $n_p \times n_p$ LEs for second derivatives. Then the conjugate gradient algorithm leads to local optimal solutions. See that the optimal passive dynamics design as a special case means **k**=**0**.

The extended Kleinman's algorithm proposed for IPCD of unsaturated linear plants

1. Initialization (i=0): Suggest a stabilizing controller $\mathbf{k}_0$ for initial plant ($\mathbf{A}_0$, $\mathbf{B}_0$) and compute the weighing matrices $\mathbf{R}_0$ and $\mathbf{Q}_0$.
2. Compute the cost by solving LE (9).
3. $\mathbf{A}_{c,i}^{T}\mathbf{P}_{i}+\mathbf{P}_{i}\mathbf{A}_{c,i}+\mathbf{Q}_{i}+\mathbf{k}_{i}^{T}\mathbf{R}_{i}\mathbf{k}_{i}=\mathbf{0}$.
4. Using parameter update law(12), improve the plant as,

$$\mathbf{p}^{i+1}=\mathbf{p}^{i}-\alpha\left[m_{rs}+\mathbf{x}_{0}^{T}\mathbf{P}_{rs}\mathbf{x}_{0}\right]^{-1}\left[m_{r}+\mathbf{x}_{0}^{T}\mathbf{P}_{r}\mathbf{x}_{0}\right],$$

in which $\mathbf{P}_r$ can be found by solving the LE (18) and $\mathbf{P}_{rs}$ can be found by solving the LE (19). α should be found so that the solution of

$$\left(\mathbf{A}_{i+1}-\mathbf{B}_{i+1}\mathbf{k}_i\right)^T \mathbf{P}_i'+\mathbf{P}_i'\left(\mathbf{A}_{i+1}-\mathbf{B}_{i+1}\mathbf{k}_i\right)+\mathbf{Q}_{i+1}+\mathbf{k}_i^T\mathbf{R}_{i+1}\mathbf{k}_i=\mathbf{0} \tag{20}$$

satisfies the inequality $\mathbf{P}_i \geq \mathbf{P}_i' \geq 0$.

5. Update the matrix by optimal policy again,

$$\mathbf{k}_{i+1}=\mathbf{R}_{i+1}^{-1}\mathbf{B}_{i+1}^T\mathbf{P}_i' \tag{21}$$

5. If $\left\|\mathbf{k}_i-\mathbf{k}_{i+1}\right\|<\varepsilon$ then end, else *i*→*i*+1 and goto 2

2.4.1. Stability barrier function

In every iteration of the above Newton iteration, the stability of the closed loop must be ensured. The stability condition can be stated as,

$$\max\mathrm{Re}\,\lambda\left(\mathbf{A}-\mathbf{B}\mathbf{k}\right)+\delta<0 \tag{22}$$

where δ is a small value to construct a boundary of exponential decay rate. This condition can be introduced as a barrier penalty function to the optimization process as,

$$J_+=\exp\left\{a\left[\max\mathrm{Re}\,\lambda\left(\mathbf{A}-\mathbf{B}\mathbf{k}\right)+\delta\right]\right\} \tag{23}$$

where a is a big number to have a rough penalty. It can be optimized by simple gradient descent algorithm and it adds a following alteration to Eq. (12) as,

$$p_i=p_i-\beta\frac{\partial J_+}{\partial p_i} \tag{24}$$

where $\frac{\partial J_+}{\partial p_i}=a\frac{\partial\Re\lambda_j}{\partial p_i}e^{a\left[\Re\lambda_j-\delta\right]}=\frac{a}{2}\left(\frac{\partial\lambda_j}{\partial p_i}+\frac{\partial\lambda_j^*}{\partial p_i}\right)e^{a\left[\Re\lambda_j-\delta\right]}$. To find the term $\frac{\partial\lambda_j}{\partial p_i}$ where j:=argmax(Reλ^T) and it can be written as [5],

$$\frac{\partial\lambda_j}{\partial p_i}=\frac{\mathrm{y}_j^T\mathbf{A}_{c,i}\,\mathrm{x}_j}{\mathrm{y}_j^T\,\mathrm{x}_j} \tag{25}$$

where x and y are right and left eigenvectors of $\mathbf{A}_c$, closed loop gain matrix defined in Eq. (14) and $\mathbf{A}_{c,i}$ is defined in (18). The term doesn't let to get into the unstable area (left half plane).

2.5. Proof of convergence

The following theorem is intended to prove the convergence.

Theorem 1. Iteration on the plant and the controller is uniformly convergent if:

(i) The modified plant with controller $\mathbf{k}_i$ is still stable.

(ii) The plant modification does not increase the cost.

Proof. We define a Lyapunov product as $\mathbf{A}*\mathbf{B}=\mathbf{A}^T\mathbf{B}+\mathbf{B}^T\mathbf{A}$. The condition (i) is equivalent to the existence of a matrix **P'** which satisfies the LE (20). Also, from condition (ii), we have $\mathbf{x}_0^T\mathbf{P}_i\mathbf{x}_0-\mathbf{x}_0^T\mathbf{P}_i'\mathbf{x}_0\geq 0$ so,

$$\mathbf{P}_i - \mathbf{P}'_i \geq 0 \tag{26}$$

Then assuming condition (i) and Eq. (26), it is possible to find a matrix as **Q'≥0** where,

$$\left(\mathbf{A}_{i+1} - \mathbf{B}_{i+1}\mathbf{k}_i\right) * \left(\mathbf{P}_i - \mathbf{P}'_i\right) = -\mathbf{Q}' \leq \mathbf{0} \tag{27}$$

Now consider the Lyapunov function $V = \mathbf{x}^T \mathbf{P}'_i \mathbf{x}$. Derivation along the plant/controller perturbed trajectory as considering of condition (i),

$$\dot{V} = \mathbf{x}^T \left[\left(\mathbf{A}_{i+1} - \mathbf{B}_{i+1}\mathbf{k}_{i+1}\right) * \mathbf{P}'_i\right]\mathbf{x} \tag{28}$$

Using Eq. (A2.4), it can be rewritten as,

$$\dot{V} = \mathbf{x}^T \left[\left(\mathbf{A}_{i+1} - \mathbf{B}_{i+1}\mathbf{k}_i\right) * \mathbf{P}'_i + \left(\mathbf{B}_{i+1}\left(\mathbf{k}_i - \mathbf{k}_{i+1}\right)\right) * \mathbf{P}'_i\right]\mathbf{x} \tag{29}$$

Using Eq. (20) it is possible to write,

$$\dot{V} = \mathbf{x}^T \left[-\mathbf{Q}_{i+1} - \mathbf{k}_i^T \mathbf{R}_{i+1}\mathbf{k}_i + \left(\mathbf{k}_i - \mathbf{k}_{i+1}\right) * \left(\mathbf{B}_{i+1}^T \mathbf{P}'_i\right)\right]\mathbf{x} \tag{30}$$

Now by substituting the update law for the controller in Eq. (21) and expanding it, say,

$$\begin{aligned}\dot{V} &= \mathbf{x}^T \left[-\mathbf{Q}_{i+1} - \mathbf{k}_i^T \mathbf{R}_{i+1}\mathbf{k}_i + \left(\mathbf{k}_i - \mathbf{k}_{i+1}\right) * \left(\mathbf{R}_{i+1}\mathbf{k}_{i+1}\right)\right]\mathbf{x} \\ &= \mathbf{x}^T \left[-\mathbf{Q}_{i+1} - \mathbf{k}_i^T \mathbf{R}_{i+1}\mathbf{k}_i + \mathbf{k}_i^T \mathbf{R}_{i+1}\mathbf{k}_{i+1} - 2\mathbf{k}_{i+1}^T \mathbf{R}_{i+1}\mathbf{k}_{i+1} + \mathbf{k}_{i+1}^T \mathbf{R}_{i+1}\mathbf{k}_i\right]\mathbf{x} \\ &= \mathbf{x}^T \left[-\mathbf{Q}_{i+1} - \left(\mathbf{k}_i - \mathbf{k}_{i+1}\right)^T \mathbf{R}_{i+1}\left(\mathbf{k}_i - \mathbf{k}_{i+1}\right) - \mathbf{k}_{i+1}^T \mathbf{R}_{i+1}\mathbf{k}_{i+1}\right]\mathbf{x} \leq 0.\end{aligned} \tag{31}$$

Therefore, if the modified system remains stable, then the controller modification is monotonically decreasing. Now, we prove that condition (ii) is sufficient to have monotonic plant convergence. To see this, suppose the following Lyapunov function $V = \mathbf{x}^T \mathbf{P}_i \mathbf{x}$. In the modified plant trajectory with controller $\mathbf{k}_i$ it can be differentiated as,

$$\begin{aligned}\dot{V} &= \mathbf{x}^T \left[\left(\mathbf{A}_{i+1} - \mathbf{B}_{i+1}\mathbf{k}_i\right) * \mathbf{P}_i\right]\mathbf{x}, \\ &= \mathbf{x}^T \left[\left(\mathbf{A}_{i+1} - \mathbf{B}_{i+1}\mathbf{k}_i\right) * \mathbf{P}_i - \left(\mathbf{A}_{i+1} - \mathbf{B}_{i+1}\mathbf{k}_i\right) * \mathbf{P}'_i + \left(\mathbf{A}_{i+1} - \mathbf{B}_{i+1}\mathbf{k}_i\right) * \mathbf{P}'_i\right]\mathbf{x}, \\ &= \mathbf{x}^T \left[\left(\mathbf{A}_{i+1} - \mathbf{B}_{i+1}\mathbf{k}_i\right) * \left(\mathbf{P}_i - \mathbf{P}'_i\right) + \left(\mathbf{A}_{i+1} - \mathbf{B}_{i+1}\mathbf{k}_i\right) * \mathbf{P}'_i\right]\mathbf{x}.\end{aligned} \tag{32}$$

Using Eq. (27), it can be written as,

$$\dot{V} = \mathbf{x}^T \left[-\mathbf{Q}' - \mathbf{Q}_{i+1} - \mathbf{k}_i^T \mathbf{R}_{i+1}\mathbf{k}_i\right]\mathbf{x} \leq 0. \tag{33}$$

This completes the proof.

3. Optimal active/passive suspension design

3.1. Mathematical modelling of the system

Considering the heave motion of the body only, it should have one DOF. An active suspension control system model is shown in Fig. 1. The suspension system can be sketched as a 2 DOF rectilinear mass-spring-damper system. The aim is minimization of body movement when the abrupt input to the system is applied due to the vertical irregularities along the road as *w*. This can be done by selecting appropriate controller $u(t)$, mainspring stiffness k_2 and dashpot's strength c_2.

State vector is selected as $\mathbf{x}=[x_1\text{-w}\ \ x_2\text{-w}\ \ \dot{x}_1\ \ \dot{x}_2]$. The control input (u) is applied by usual pneumatic jack between the body of the automobile and the tires. The dynamic equation is:

$$\mathbf{A}=\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -\dfrac{k_1+k_2}{m_1} & \dfrac{k_2}{m_1} & -\dfrac{c_1+c_2}{m_1} & \dfrac{c_2}{m_1} \\ \dfrac{k_2}{m_2} & -\dfrac{k_2}{m_2} & \dfrac{c_2}{m_2} & -\dfrac{c_2}{m_2} \end{bmatrix},\mathbf{B}=\begin{bmatrix} 0 \\ 0 \\ -\dfrac{1}{m_1} \\ \dfrac{1}{m_2} \end{bmatrix} \tag{34}$$

where k_2 and c_2 are mainspring stiffness and damping coefficient of the suspension system and they are passive unknowns, m_2 denotes the body mass, m_1, (c_1=0) and k_1 are wheel parameters.

The numerical values here is included in Table 1.

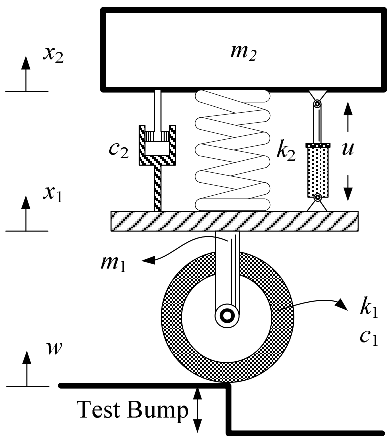

Fig. 1: Case study: Active suspension system

Table 1. Known passive parameters with no ability to design

Parameter	Values
Sprung mass(m_2)	288.9(kg)
unsprung mass(m_1)	28.58(kg)
Tire stiffness(k_1)	1.559×10^5(N/m)

3.2. Simulations

In this section, the optimization of the above suspension system is applied for the initial condition as an abrupt input in w as $\mathbf{x}_0=[-0.1\ -0.1\ 0\ 0]^T$. Full accurate state feedback is considered here and it means there is no problem with observation and measurement.

To constrain the parameters to positive values we consider their squares in the system's dynamics as $k_2 = p_1^2, c_2 = p_2^2$. Then, by using exhaustive solving of the series LQR problems in the whole of the parameter space (p_1, p_2), Fig. 2 is generated. In this figure, it is shown that the optimal value of the parameters is located at (p_1,p_2)=(230.48,72.66) corresponding to (k_1,c_1)=(53121,5279.5). Therefore, it is sufficient to show that the algorithm proposed above will converge to these parameters. To check the stability max($\Re(\lambda_{max})$) is shown in the Fig. 3.

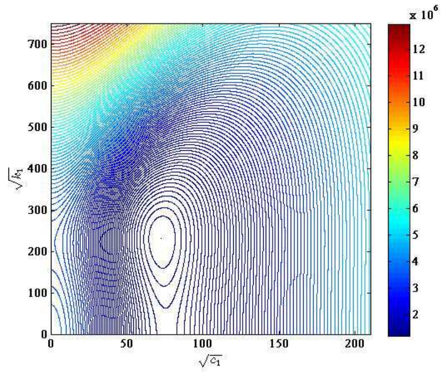

Fig. 2: Cost map on the design parameters

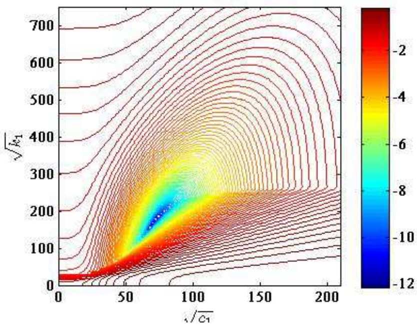

Fig. 3: Max-real of eigenvalue for closed loop

The convergence of the passive and active parameters are shown in Fig. 4 and Fig. 5. In Fig. 6, the cost minimization is shown. In Fig. 7 parameter convergence is shown for contours of the cost map and it can be seen that the algorithm is rapidly converged to the their optimal values. In comparison to salama's iteration, the proposed method run in 2.3s where the salama iteration was run in 12s. Thereupon, an improvement of 81% is obtained in the solution time. Note that the results are identical.

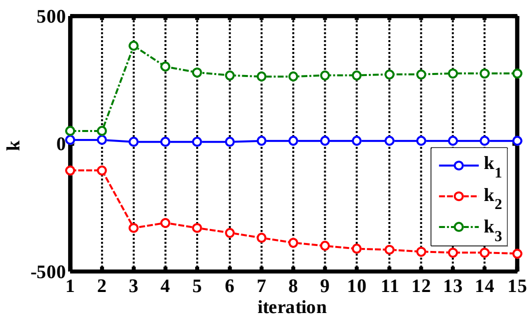

Fig. 4: Gain convergence

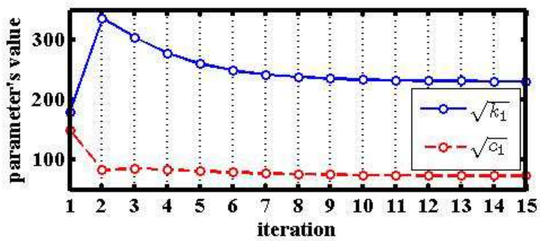

Fig. 5: Parameter convergence

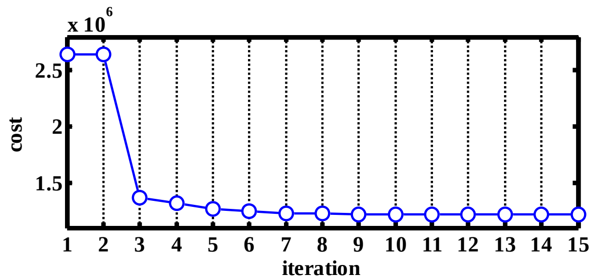

Fig. 6: Cost convergence

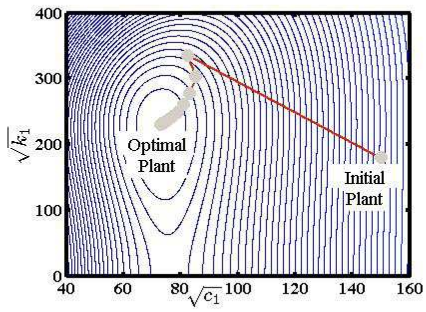

Fig. 7: Convergence in the cost map

For the optimized system, the initial response is shown in the following figures.

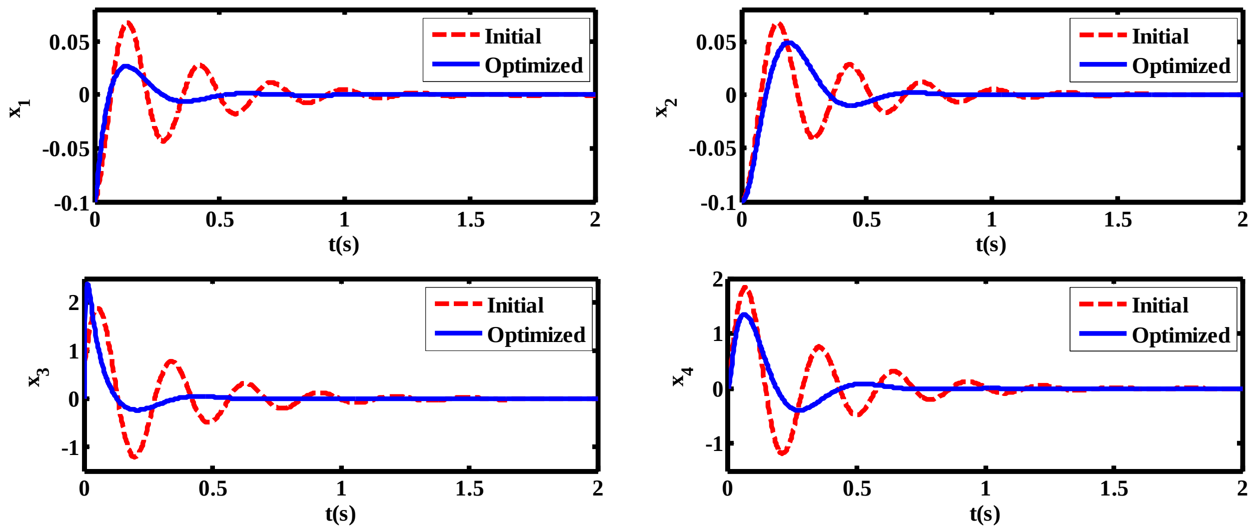

Fig. 1: States of the system in normal/optimal cases

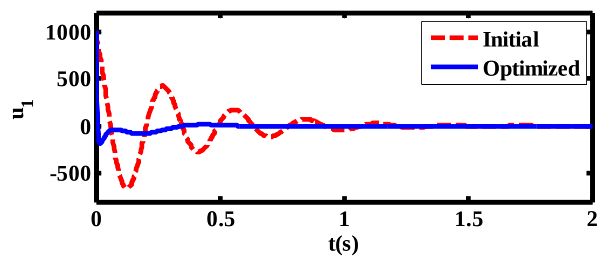

Fig. 2: Comparing of control signals

4. Conclusion

In this article, the well-known Kleinman's [8] algorithm for solution of the ARE based on the Newton algorithm is extended. Then, it is proposed to find an appropriate passive parameters set integrated with the controller. This IPAD problem solving methodology is applied to a suspension system. The method has a straightforward scheme intentionally. Its efficiency is shown by simulations. The future work includes the extension of this method to optimal control of bilinear systems that appeared in the semi-active suspension systems with variable damping coefficient. The LEs can be solved for large scale systems with accessible packages such as MATLAB® and this method can be applied on the integrated active/passive design of trusses and flexible systems such as shape design problems.

References

[1] J.T. Allison, D.R. Herber, Multidisciplinary design optimization of dynamic engineering systems, AIAA Journal, 52 (2014) 691-710.

[2] A.L. Hale, W.E. Dahl, J. Lisowski, Optimal simultaneous structural and control design of maneuvering flexible spacecraft, Journal of Guidance, Control, and Dynamics, 8 (1985) 86-93.

[3] M. Salama, J. Garba, L. Demsetz, F. Udwadia, Simultaneous optimization of controlled structures, Computational mechanics, 3 (1988) 275-282.

[4] R.T. Haftka, Integrated structure-control optimization of space structures, in: AIAA/ASME/ASCE/AHS 30th Structures, Structural Dynamics, and Materials Conference, Long Beach, CA, 1990.
[5] D.S. Bodden, J.L. Junkins, Eigenvalue optimization algorithms for structure/controller design iterations, Journal of Guidance, Control, and Dynamics, 8 (1985) 697-706.
[6] R.A. Canfield, L. Meirovitch, Integrated structural design and vibration suppression using independent modal space control, AIAA journal, 32 (1994) 2053-2060.
[7] J.T. Allison, Z. Han, Co-design of an active suspension using simultaneous dynamic optimization, in: ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Washington, DC, USA, 2011.
[8] D. Kleinman, On an iterative technique for Riccati equation computations, IEEE Transactions on Automatic Control, 13 (1968) 114-115.
[9] N.R. Sandell Jr., On Newton's method for Riccati equation solution, IEEE Transactions on Automatic Control, 19 (1974) 254-255.