www.elsevier.com/locate/ress

Planning structural inspection and maintenance policies via dynamic programming and Markov processes. Part II: POMDP implementation

K.G. Papakonstantinou, M. Shinozuka

PII: S0951-8320(14)00068-4
DOI: http://dx.doi.org/10.1016/j.ress.2014.04.006
Reference: RESS5026

To appear in: *Reliability Engineering and System Safety*

Planning structural inspection and maintenance policies via dynamic programming and Markov processes. Part II: POMDP implementation

K.G. Papakonstantinou*, M. Shinozuka

Department of Civil and Environmental Engineering, University of California Irvine, Irvine, USA

* Corresponding author. Tel.: +1 949 228 8986
Email address: kpapakon@uci.edu

Abstract

The overall objective of this two part study is to highlight the advanced attributes, capabilities and use of stochastic control techniques, and especially Partially Observable Markov Decision Processes (POMDPs), that can address the conundrum of planning optimum inspection/monitoring and maintenance policies based on stochastic models and uncertain structural data in real time. In this second part of the study a distinct, advanced, infinite horizon POMDP formulation with 332 states is cast and solved, related to a corroding reinforced concrete structure and its minimum life-cycle cost. The formation and solution of the problem modernize and extend relevant approaches and motivate use of POMDP methods in challenging practical applications. Apart from uncertain observations the presented framework can also support uncertain action outcomes, non-periodic inspections and choice availability of inspection/monitoring types and intervals, as well as maintenance actions and action times. It is thus no surprise that the estimated optimum policy consists of a complex combination of a variety of actions, which cannot be achieved by any other method. To be able to solve the problem we resort to a point-based value iteration solver and we evaluate its performance and solution quality for this type of applications. Simpler approximate solvers based on MDPs are also used and compared and the important notions of observation gathering actions and the value of information are briefly discussed.

Keywords: Partially Observable Markov Decision Processes, optimal stochastic control, belief space, uncertain observations, structural life-cycle cost, infrastructure management.

1. Introduction

This paper complements a companion paper, [1], that provided a thorough theoretical background on the use of Markov Decision Processes (MDPs) for infrastructure management. As shown in our Part I companion paper the Partially Observable case of Markov Decision Processes (POMDPs) provides an excellent choice for decision making and asset management under uncertainty, with firm mathematical foundations and superior attributes. POMDPs are capable of describing a huge number of realistic situations and can support a diverse range of formulations and objective functions, including condition-based, reliability and/or risk-based problems, [1]. The optimum life-cycle policies are provided based on stochastic control, probabilistic models, uncertain structural data and Bayesian principles. This Part II paper emphasizes more on the details of POMDP models and solvers for optimum structural inspection and maintenance policies. A detailed application example for a corroding reinforced concrete structure is provided and among others the cost-benefit of information is naturally incorporated in the formulation. Most quantities and notions already defined in Part I, [1], are not defined again in detail in this work and any assumptions made in the companion paper are valid in this paper as well. For example, in this paper we will again only refer to rewards, since cost can be simply perceived as negative rewards.

Markov Decision Processes have a long, successful history of implementation in risk management and minimum life-cycle costing of civil engineering structures, [2]. Perhaps the strongest indication of their success and capabilities is their use from different state agencies all over the world for asset management of a variety of infrastructures, like bridges, transportation networks, pavements, etc., [3-5]. In United States, PONTIS, the predominant management system for bridges and other infrastructures, uses MDPs as its core optimization tool, [6-8]. PONTIS is currently a registered trademark of AASHTO and it is licensed and used by the majority of U.S. state transportation departments and other organizations in the U.S. and other countries.

Although MDPs provide a very strong and versatile mathematical framework for asset management they also have some limitations which, at certain occasions, may be crucial for the quality of solutions they provide. POMDPs are a much more general tool that inherit all the valuable attributes of MDPs and add more. However, POMDPs comprise a newer scientific field, open to extensive scientific research currently and not as mature as the MDP one. These reasons, in addition to the fact that are much harder to be solved adequately for large, complex, realistic problems, has led until now to very few works addressing them in the context of optimum inspection and maintenance, in comparison to other approaches in this area, [1]. In Madanat & Ben-Akiva [9] a POMDP problem with 8 states and a finite horizon of 10 years is solved and in Smilowitz & Madanat [10] a problem of just 3 states, concerning a network of highway pavements, is presented. Both of these works use a fixed, regular grid and the nearest neighbor interpolation-extrapolation rule (enduring all the disadvantages of this method, discussed in the Part I paper [1]) and convert the problems into fully observable MDPs, which are then solved by dynamic and linear programming, respectively. In Ellis et al. [11] and Jiang et al. [12] some finite horizon POMDP problems are analyzed, concerning structural degradation of bridge girders due to corrosion and fatigue. The maximum size of the state spaces in these two works is 13 and the authors solve the problems with an exact algorithm, taking advantage of the small number of states and the finite horizon formulation, which in this occasion is computationally beneficial, in comparison to an infinite horizon case. Faddoul et al. [13] studied an inspection and maintenance problem, regarding a reinforced concrete highway bridge deck, and sought optimum policies based on the nearest neighbor method and a 5 state POMDP with a horizon length of 20 years. Use of POMDPs in works pertinent to the discussion herein can be further found in [14-17]. The maximum state space size used in these cases is 9, [14]. It is apparent that in all these works the POMDP formulation of the problems hindered the researchers from describing the system in a more refined way, with larger state spaces. In MDP formulations, where solutions can be much more easily found, state space sizes in the order of hundreds or thousands are commonly encountered and can be even considered small. An example for instance is the work by Robelin & Madanat [18] where a bridge deck management problem, formed as a MDP with 840 states, is solved. The state space in this case consists of the reliability index of the deck and history dependent parameters.

In this paper, we rely on these previously presented works and, efficacious approaches, as in [11] and [19], and significantly extend them towards large-scale modeling and solution of realistic problems. Utilizing the presented topics in the Part I companion paper, a distinct, advanced, demanding infinite horizon formulation is cast and solved in this work with non-stationary stochastic phenomena, connection to physically based stochastic models and a considerably larger state-space of 332 states. Choice availability of different monitoring and maintenance actions, uncertain observation and action outcomes and non-periodic structural visits are also incorporated in this work. With such an unprecedented formation and variety of unimpeded options the estimated optimum policy is a highly complex combination of a range of inspection/monitoring types and intervals, and maintenance actions and action times, which cannot be achieved by any other method. To be able to solve this challenging problem that could not have been solved by techniques in the aforementioned references, we resort to point-based solvers, as explained in our companion paper, [1]. Point-based methods have been mainly developed in the field of artificial intelligence for autonomous robot navigation which is a problem with inherently different characteristics from the structural management problem. Among others, uncertainty in usual robot navigation problems decreases with time, since the terrain is gradually explored, and not the opposite like in structural maintenance, and different observation actions are not typically sought during planning. Despite the differences, we demonstrate in this work that the point-based value iteration algorithm Perseus, [20], can perform successfully in this type of applications, even for difficult problems with larger state-spaces than the ones currently described in the maintenance literature. Additional, recent attempts by the authors with larger models, with thousands of states, in a finite horizon formulation can be seen in [21], where the POMDP mapping to physically based stochastic models is also explained in greater detail. Apart from Perseus we also solve the problem with simple approximate solvers (MLS, QMDP, [1]) that are directly based on MDPs. Current structural management systems (like PONTIS) only rely on MDPs and hence they could straightforwardly utilize these methods. We demonstrate differences in performance and quality of solution between these methods and Perseus and based on this comparison we also shed some extra light to the important notions of observation gathering actions and the value of information, which of course implies that more accurate and precise inspection/monitoring techniques are self-evidently more expensive than cruder inspection/monitoring methods. Compendiously, the current paper provides in detail a generic framework that modernizes the way relevant problems are solved today, sets a step forward in large-scale modeling, showcases deployment of point-based POMDP methods and motivates their use in a wider variety of problems and practical applications.

The particular application used in this work, in order to demonstrate the suggested POMDP framework, its detailed implementation, solution and attributes, relates to a corroding reinforced concrete structure. Unfortunately, current non-destructive corrosion evaluation techniques are prone to measurement errors and have inherent deficiencies, which make it difficult to derive, certain, reliable engineering conclusions based on their output, [22-24]. For this reason, a POMDP formulation of the problem is most appropriate. A spatial, stochastic, physically based model of steel corrosion in a wharf deck reinforced concrete slab is developed in Papakonstantinou & Shinozuka [25]. Based on this modeling an infinite horizon, non-stationary POMDP model with yearly time-steps and 332 discrete physical states is cast in this work and solved by asynchronous dynamic programming and Perseus, [20]. The objective of the application is to identify an optimum life-cycle cost policy that can suggest, without any modeling restrictions, when and what type of inspection/monitoring and maintenance actions should be employed based on the conditions of the deteriorating structure in real time [1]. On the whole, 4 different maintenance actions are considered (including doing nothing and full replacement of the structure) and 3 different inspection/monitoring actions (including no inspection), resulting in a total number of 10 considered, different, combined (maintenance – inspection) action choices for the decision-maker. The uncertain observation outcomes are categorized in 4 different possible conditions and inform the decision-maker about the structural status according to the accuracy and precision of the chosen evaluation method. In general, the most important characteristic of all the actions for the optimum life-cycle cost policy is their relative to each other effectiveness and cost, which are explained in detail in the paper. In the remainder of this work, we thoroughly present the POMDP modeling of the problem, emphasizing mostly in the do-nothing action, which of course relates to the deterioration process. The rewards part of the modeling and the combination of maintenance-inspection actions are described as well. In section 3 the point-based value iteration algorithm, Perseus, is analyzed, while in section 4 specific implementation details and comprehensive results are provided and discussed.

2. POMDP modeling

POMDP is a 6-tuple (*S, A, P, O, Po, R*) where, *S*, *A* and *O* finite set of states, actions and possible observations respectively, *P* state transition probabilities, *Po* observation probabilities modeling the effect of actions and states on observations, and *R* rewards. All these parameters constitute the input of the model and have to be defined accordingly.

2.1. *State transition probabilities for deteriorating structures*

Finding state transition probability matrices for deteriorating structures is not an easy task in practice, mainly due to lack of reliable, historical field data. Several different ways can be found in the literature addressing this issue, based on time-series observations of facility performance states, expert's judgment or analytical/numerical models. In PONTIS [6] a combination of expert's judgment with historical data is used and transition matrices are refined accordingly whenever new data become available. Different regression approaches and techniques in order to calculate transition matrices based on field data can be seen in [26-29], while in [12,18] numerical experiments are used.

Based on the probability vector, $\mathbf{z_t}(s)$, of being in any state of the system, in time t, regardless of how this information has been obtained (i.e. numerical data, field observations, etc.), the transition matrices **P** may be found, in certain occasions, using the basic equation:

$$\mathbf{z}_t = \mathbf{z}_{t-1} \bullet \mathbf{P}_t \quad (1)$$

Another, more general, but data demanding method, is to directly calculate the transition probabilities by:

$$p(s'|s) = \frac{\text{number of transitions from } s \text{ to } s'}{\text{number of data in state } s}, \quad (2)$$

or, even more generally, by hidden Markov model techniques. In many circumstances field data do not support the Markovian or lack-of-memory property, e.g. [29], and thus non-stationary transition matrices are found, e.g. [30]. In these cases the probability density function of duration time in some state of the system, if known, can also be used to form the transition matrices, as also shown in the companion paper.

Fitting transition probability matrices based on historical field data alone requires many observations that are rarely available in reality, for identical structures under similar deterioration conditions. Unavoidably, samples from a non-homogeneous group of structures is often selected, to acquire a statistically acceptable number of data, and the difficulty is then rolled over to the formulation of the regression model, correct choice of covariates, etc. To make things even more complicated, structures with dissimilar maintenance history should not normally be included in the same statistical group and the collection process and archiving of reliable field data is currently, in most cases, very recent and long term performance of structures is missing. To handle these issues, the physically based stochastic corrosion model presented in detail in [25] is utilized in this work, in order to formulate the corrosion deterioration transition probability matrices. By having a numerical model of deterioration, which can be supported and updated by field data, many complications of non-model based approaches, relying on regression methods, are avoided.

2.1.1. *Non-stationary deterioration transition*

Following AASHTO [31] the steel corrosion in the 24m by 15.2m wharf deck reinforced concrete slab in [25] can be characterized by 4 discrete *conditions*. In condition 1 the spatial extent of damage due to corrosion is less than 10%, in conditions 2 and 3 it is between 10%-25% and 25%-50% respectively and finally in condition 4 the extent of corrosion is over 50%. Results from the continuous stochastic modeling revealed that the deterioration phenomenon does not possess the Markovian property of independence from history. The same conclusion, based on field data, for deterioration events governed by chemical processes can be found in [29]. Consequently, we consider that the *deterioration rate* is different and generally more adverse with every year that passes without maintenance and the state space of the problem, for the infinite horizon case, consists of the *condition* of the structure in each different *deterioration rate*. Based on the state augmentation technique, presented in our Part I paper, the combination of structure *conditions* and different *deterioration rates* results in 332 different total states for this problem, as will be also further explained shortly. Adopting the very common assumption for this type of problems that the structure can deteriorate only by one condition, utmost, during each year, [1], which was also supported by the continuous model results, one representative part of the 332 by 332 transition probability matrix **P** for the do-nothing action is formatted as:

$$\mathbf{P}_{(332x332)} = \begin{bmatrix} \ddots & \vdots & \iddots \\ \cdots & \mathbf{P}(a \to b) & \cdots \\ \iddots & \vdots & \ddots \end{bmatrix}$$

$$\mathbf{P}(a \to b)_{(8x8)} = \begin{bmatrix} \mathbf{0}^{aa}_{(4x4)} & \begin{matrix} p_{11}^{ab} & p_{12}^{ab} & 0 & 0 \\ 0 & p_{22}^{ab} & p_{23}^{ab} & 0 \\ 0 & 0 & p_{33}^{ab} & p_{34}^{ab} \\ 0 & 0 & 0 & p_{44}^{ab} = 1 \end{matrix} \\ \mathbf{0}^{ba}_{(4x4)} & \mathbf{0}^{bb}_{(4x4)} \end{bmatrix}, \tag{3}$$

where a and b represent arbitrary, consecutive deterioration rates and the numbering represents the 4 slab conditions mentioned earlier. Thus, the $\mathbf{P}(a \to b)$ submatrix shows that without any maintenance the only non-zero probabilities are the ones that describe the transition of the structure from deterioration rate a to deterioration rate b and from condition x to condition x or x+1. In case the structure is already in condition x=4 (extent of corrosion over 50%), it will remain in the same condition x=4 with probability 1, if no maintenance is performed. Similarly, a $\mathbf{P}(b \to c)$ submatrix would describe the transition of the structure from deterioration rate b to the subsequent deterioration rate c and from condition x to condition x or x+1.

All parameters of the deterioration matrix were calculated based on Eq. (2) and the physically based stochastic model in [25] with yearly intervals, a total time duration of 100 years and 1000 Monte Carlo model simulations starting from a no damage condition. Further details about this procedure can be seen in [21]. In Fig. 1, the probabilities of being in any condition, starting from perfect condition, calculated based on Eqs. (1) and (2) are compared against the continuous stochastic modeling results. The results are in very close agreement and this proves the adequacy of the discrete Markov chain formulation, in this application. The physically based stochastic modeling results are shown in Fig. 1 both unsmoothed, simply derived from the Monte Carlo samples and standard empirical cumulative density function calculations, and smoothed, through a standard nonparametric kernel density estimator with a Gaussian kernel.

The deterioration transition matrix could have also been estimated based on semi-Markov processes principles, as explained in the companion Part I paper, or even Eq. (1) in this case. However, since in this application closed form expressions for the functions needed in these approaches were not available and the mapping from the continuous stochastic model to the discrete POMDP model is based on Monte Carlo simulations, the advantage of using these approaches over Eq. (2) is somewhat lost. Anyhow, the acquired mapping with the used method is excellent.

Based on the choice of the 4 AASHTO conditions and the used physically based model, there are no condition transitions from condition 1 until the deterioration rate equals 18, (Fig. 1). In other words, the considered structure cannot be damaged by corrosion by more than 10% of its extent in the first 18 years after its construction. Therefore, the deterioration rates from 1 to 17 are alike and can be combined in one deterioration rate for the POMDP modeling. As a result, and given that the simulated data from the physically based modeling consider up to 100 years since the initial construction, the infinite horizon POMDP case is described by 332 total states, consisting of combinations of 83 (100-17) different deterioration rates and 4 slab conditions in each rate.

As also indicated in Fig. 1, none of the simulated 1000 Monte Carlo samples entered condition state 3 earlier than 30 or 20 years or condition state 2 earlier than, let's say 5 years, and thus there are no evidences of how would the system behave in such occasions. It is therefore assumed that it would remain in the same condition state with certainty, until a deterioration rate is reached at which its behavior has been documented. Provided that the physical based modeling represents reality adequately this assumption does not cause significant modeling errors, unless pathogenic cases are of interest; such as having deterioration condition level 3 at the first deterioration rate, etc. Finally, if the system reaches the last simulated deterioration rate it is assumed that it will continue to deteriorate according to that rate level, since subsequent, possibly more severe, rates are currently undocumented. The interested reader can observe in detail all the necessary parameters that form the very sparse 332 by 332 deterioration transition matrix in [21].

2.2. *Remaining POMDP parameters (actions, observations, rewards)*

Until now in this paper, the state space of the problem and the deterioration transition matrices for the do-nothing action have been presented. The remaining action transition matrices, observation matrices and rewards considered in this work do not follow the same detailed level just presented for the do-nothing action and although representative, for the considered problem, are nonetheless arbitrary and mostly serve the mathematical requirements of the formulation.

For the current example, apart from the do-nothing action, three more maintenance actions are considered. The easiest to describe is the replacement action. If the decision-maker chooses to replace the structure, no matter the state in which the structure is at that time, it will transition with certainty to the state with condition level 1 and the first deterioration rate.

Another maintenance action that is considered is a minor-repair action. If a minor-repair action is performed, the system still transitions to the next deterioration rate as in the do-nothing action but the condition level of the structure possibly improves or at worst remains the same. The core minor-repair matrix, which repeats itself many times in the global, 332 by 332 minor-repair transition matrix following the just explained pattern, is given by:

$$\mathbf{P}(a \rightarrow b)_{(8x8)} = \begin{bmatrix} \mathbf{0}^{aa}_{(4x4)} & \begin{matrix} p^{ab}_{11} = 1.00 & 0 & 0 & 0 \\ p^{ab}_{21} = 0.67 & p^{ab}_{22} = 0.33 & 0 & 0 \\ p^{ab}_{31} = 0.35 & p^{ab}_{32} = 0.48 & p^{ab}_{33} = 0.17 & 0 \\ p^{ab}_{41} = 0.14 & p^{ab}_{42} = 0.42 & p^{ab}_{43} = 0.36 & p^{ab}_{44} = 0.08 \end{matrix} \\ \mathbf{0}^{ba}_{(4x4)} & \mathbf{0}^{bb}_{(4x4)} \end{bmatrix}, \quad (4)$$

and the symbols follow the same definitions as in Eq. (3). In a physical context, the minor-repair action can represent patch repairing of the concrete slab at possible contaminated locations, [32].

The last maintenance action considered is a major-repair action. If a major-repair action is performed, the system transitions with certainty at the next epoch to three deterioration rates before the one currently in and the condition level of the structure possibly improves or at worst remains the same. The core major-repair matrix, which repeats itself many times in the global, major-repair transition matrix following the just explained pattern, is given by:

$$\mathbf{P}(a \leftarrow d)_{(8x16)} = \begin{bmatrix} \mathbf{0}^{ca}_{(4x4)} & \mathbf{0}^{c[bcd]}_{(4x12)} \\ \begin{matrix} p^{da}_{11} = 1.00 & 0 & 0 & 0 \\ p^{da}_{21} = 0.89 & p^{da}_{22} = 0.11 & 0 & 0 \\ p^{da}_{31} = 0.73 & p^{da}_{32} = 0.24 & p^{da}_{33} = 0.03 & 0 \\ p^{da}_{41} = 0.56 & p^{da}_{42} = 0.34 & p^{da}_{43} = 0.09 & p^{da}_{44} = 0.01 \end{matrix} & \mathbf{0}^{d[bcd]}_{(4x12)} \end{bmatrix}, \quad (5)$$

where *a*, *b, c, d* represent consecutive deterioration rates and the remaining symbols follow the same definitions as in Eq. (3). Obviously, if the system is in deterioration rates from 1 to 4 it will go to a state with deterioration rate 1 in the next time step. In a physical context, the major-repair action can represent use of waterproofing membranes or concrete milling and overlays at possible contaminated locations, [32].

Having introduced the maintenance actions and their representation in the POMDP context, it is easily understood that the actions, apart from the do-nothing action, although representative, are based on synthetic transition matrices. While it is always possible to form these matrices in an equivalent manner as the do-nothing action, or maybe calibrate them from field data, we have not done these for this work. From our literature review on maintenance actions however, it was concluded that this is an area that has not received much attention. Analytical\numerical studies based on modeling and simulation of maintenance actions, e.g. [33], or empirical\experimental works based on field data are not that many and further studies on the very important subject of maintenance actions should be performed in the future.

Concerning observations in the POMDP formulation, three different assessment techniques are considered and representatively modeled. In this example, it was assumed that the state of deterioration rate of the structure is known to the decision-maker and the uncertain inspections reveal information over its condition level. The dimensions of the observation matrices are *number of states* by *number of observations* and 4 observations are considered in this problem, one for each condition level of the slab.

The first observation choice considered is the no-inspection one and in the POMDP context it is modeled through the observation matrix:

$$\mathbf{P_o}_{(332x4)} = \begin{bmatrix} p_{11}^a = 1 & 0 & 0 & 0 \\ p_{21}^a = 1 & 0 & 0 & 0 \\ p_{31}^a = 1 & 0 & 0 & 0 \\ p_{41}^a = 1 & 0 & 0 & 0 \\ p_{11}^b = 1 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ p_{41}^b = 1 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ & & & \end{bmatrix}, \qquad (6)$$

where p_{11}^a now represents probability of being in a state with deterioration rate *a* and condition level 1 and observing the first observation. The remaining parameters are similarly explained. By a close look at Bayes' rule:

$$b(s') = \frac{p(o|s',\alpha)}{p(o|\mathbf{b},\alpha)} \sum_{s \in S} p(s'|s,\alpha) b(s), \qquad (7)$$

where *b* belief, α performed action, *o* observation and $p(o|\mathbf{b},\alpha)$ the usual normalizing constant, [1], one can conclude that the specific observation matrix in Eq. (6) does not update the belief of the decision-maker because the likelihood function $p(o|s',\alpha)$ gives the exact same information no matter what the state of the structure is. An alternative way to form this matrix could have been to assume that in any state of the system the decision-maker has equal probabilities of observing any out of the 4 condition levels of the structure. This formulation however results unnecessarily in full matrices, instead of the sparse one in Eq. (6) and hinders the computational performance of Perseus.

The second inspection choice is assumed to be a visual inspection (without any loss of generality concerning evaluation techniques and/or monitoring methods). The observation matrix in this case is modeled as:

$$\mathbf{P_o}_{(332x4)} = \begin{bmatrix} p_{11}^a = 0.63 & p_{12}^a = 0.37 & p_{13}^a = 0.00 & p_{14}^a = 0.00 \\ p_{21}^a = 0.10 & p_{22}^a = 0.63 & p_{23}^a = 0.27 & p_{24}^a = 0.00 \\ p_{31}^a = 0.00 & p_{32}^a = 0.10 & p_{33}^a = 0.63 & p_{34}^a = 0.27 \\ p_{41}^a = 0.00 & p_{42}^a = 0.00 & p_{43}^a = 0.20 & p_{44}^a = 0.80 \\ p_{11}^b = 0.63 & p_{12}^b = 0.37 & p_{13}^b = 0.00 & p_{14}^b = 0.00 \\ p_{21}^b = 0.10 & p_{22}^b = 0.63 & p_{23}^b = 0.27 & p_{24}^b = 0.00 \\ p_{31}^b = 0.00 & p_{32}^b = 0.10 & p_{33}^b = 0.63 & p_{34}^b = 0.27 \\ p_{41}^b = 0.00 & p_{42}^b = 0.00 & p_{43}^b = 0.20 & p_{44}^b = 0.80 \\ \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ & & & \end{bmatrix}, \quad (8)$$

and symbols follow the same definitions as in Eq. (6), e.g. p_{23}^b represents that the probability of being in the state with deterioration rate *b* and condition level 2 and observing the third observation (which is incorrect since the system is in condition level 2 and deterioration rate *b*) is equal to 0.27.

Finally, the third inspection choice is assumed to be a monitoring inspection based on some non-destructive corrosion evaluation method, [22]. The observation matrix in this case is modeled similarly to Eq. (8) and therefore its representation is simplified here:

$$\mathbf{P_o}_{(332x4)} = \begin{bmatrix} p_{11}^a = 0.80 & p_{12}^a = 0.20 & p_{13}^a = 0.00 & p_{14}^a = 0.00 \\ p_{21}^a = 0.05 & p_{22}^a = 0.80 & p_{23}^a = 0.15 & p_{24}^a = 0.00 \\ p_{31}^a = 0.00 & p_{32}^a = 0.05 & p_{33}^a = 0.80 & p_{34}^a = 0.15 \\ p_{41}^a = 0.00 & p_{42}^a = 0.00 & p_{43}^a = 0.10 & p_{44}^a = 0.90 \\ p_{11}^b = 0.80 & p_{12}^b = 0.20 & p_{13}^b = 0.00 & p_{14}^b = 0.00 \\ \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ & & & \end{bmatrix} \quad (9)$$

The improved accuracy of this inspection choice, in comparison to the visual inspection option, is apparent by simply comparing the two observation matrices in Eqs. (8) and (9).

In a more detailed effort to form the observation matrices, Probability of Detection (PoD) curves would have been needed for the non-destructive evaluation methods and studies, based on field data and inspectors' judgment, for the visual inspection. Since the problem has a spatial aspect (extent of corrosion damage), stochastic modeling of different inspection techniques is probably required in order to identify the observation matrices accurately enough (equally to the do-nothing action modeling and mapping) and perhaps a more refined discretization of the condition level of the structure to avoid extensive marginalization over large segments. These approaches have not been followed in this work and can be considered in future studies.

Not having performed meticulous analysis to form the observation matrices, the main logic in representatively forming them is that the two informative inspection techniques should have different levels of observation accuracy. It is also assumed that for corrosion detection it is more probable to mischaracterize the condition of the structure towards a poorer condition level than a better one. The observation outcome is more informative when the true state of the structure is at the middle of a condition level segment (for example being in condition 3 and having 38% extent of damage) rather than its edges (for example being in condition 3 and having 49% extent of damage). However, the belief of the decision-maker over the condition of the structure is usually weak at the edges of the condition level segments and that is the time when inspection is needed. For these reasons, the observation probabilities are assigned having the edges of the condition level segments in mind and a conservative, pessimistic outlook of their efficiency.

An important note at this point should be that self-evidently the more accurate the observation modeling in a POMDP formulation is the better policy the decision-maker will come up with. However, after solving the POMDP, and given that the policy planning is performed over the belief space, any predictable or unpredictable real observation outcome can be handled, since at the end the action to be taken is decided based on the final belief of the decision-maker. This POMDP attribute can be also extended for actions and most importantly even if the decision-maker does not follow the model's suggestions at some point in time, the policy still remains valid and based on the current belief will try to guide decisions towards optimum results.

The rewards (costs) of the POMDP modeling in arbitrary units are given in Table 1. As explained already, costs can be simply perceived as negative rewards. The values of the maintenance actions are realistically assumed to depend on the condition level of the structure and are valued according to their expected effect on the structure's condition. The replace action is valued equally in all conditions of course and the do-nothing action has zero value. The values of the inspection actions are independent of the condition of the system and the no-inspection choice has zero value. The value of information is taken into account in this modeling, since more precise inspection techniques are naturally more expensive. Finally, the user/penalty rewards (costs) represent the estimated values the decision-maker (or the asset owner) is charged each year due to the condition of the structure. What these negative rewards resemble can be interpreted in a variety of ways, like reduction in the level of service due to deterioration, working accidents, additional expenses incurred during operation, etc. They can also resemble and be correlated with more abstract notions, like environmental impacts, structural reliability, structural failure probability, competitiveness loss, risk management, etc. Overall, user/penalty rewards are a very important component of any POMDP modeling and one of the main reasons for their versatile modeling ability and successful implementation in a variety of seemingly totally unrelated problems. Given that the rewards in this POMDP modeling are independent of the deterioration rate it is easily understood that Table 1 contains all the necessary information for the full reward function representation of all states. The only parameter that is not set yet is the discount factor γ. Assuming a typical discount rate of 0.0526 the discount factor used in this work is equal to γ=0.950.

Table 1. Rewards part of the POMDP modeling.

Although in this modeling indicative values are used, these are not based on real data or invoices. However, actual values are not usually hard to find. State agencies and companies provide lists with services and costs, e.g. [34], and lots of studies on the subject can be found on a more practical level, concerning user costs [35] and overall costs [36].

The decision-maker in this application needs to find both optimum maintenance and inspection actions. For this reason, the 4 different maintenance and the 3 different inspection actions are combined into a set of 10 distinct actions. The 2 combinations that are omitted from the full factorial combinations are the replace-visual-inspection and replace-monitoring-inspection actions, because after the structure is replaced is characterized with certainty by the state with condition level 1 and the first deterioration rate. Hence, in this application the decision-maker has an initial belief and has to choose what action to take. After choosing an action, the maintenance part of that action is performed and at the next step the user reward is collected and the inspection part of the action is executed. Based then on the new belief the decision-maker has to choose another action to take and so on. The optimal value function representation V^*, [1], in this application can therefore be given in its most expanded form by:

$$\begin{aligned} V^*(\mathbf{b}) &= \\ \max_{\alpha\in A}&\left[\sum_{s\in S} b(s)\left\{r_m(s,\alpha)+\gamma r_{ins}(s,\alpha)+\gamma\sum_{s'\in S} p(s'|s,\alpha)\,r_{pen}(\alpha,s')\right\}+\right.\\ &\left.+\gamma\sum_{o\in O}\sum_{s'\in S} p(o|s',\alpha)\sum_{s\in S} p(s'|s,\alpha)\,b(s)\,V^*(b(s'))\right], \end{aligned} \tag{10}$$

where r_m, r_{ins} and r_{pen} maintenance, inspection and penalty reward part of the action α respectively. Combining all the reward parts, as:

$$r(s,\alpha,s') = r_m(s,\alpha)+\gamma r_{ins}(s,\alpha)+\gamma r_{pen}(\alpha,s'), \tag{11}$$

the usual form of the value function is retrieved:

$$\begin{aligned} V^*(\mathbf{b}) = \max_{\alpha\in A}&\left[\sum_{s\in S} b(s)\sum_{s'\in S} p(s'|s,\alpha)\,r(s,\alpha,s')+\right.\\ &\left.+\gamma\sum_{o\in O}\sum_{s'\in S} p(o|s',\alpha)\sum_{s\in S} p(s'|s,\alpha)\,b(s)\,V^*(b(s'))\right], \end{aligned} \tag{12}$$

and in a more condensed form by:

$$V^*(\mathbf{b}) = \max_{\alpha \in A}\left[\sum_{s \in S} b(s)R(s,\alpha) + \gamma \sum_{o \in O} p(o|\mathbf{b},\alpha)V^*(\mathbf{b}_{s'})\right] \qquad (13)$$

Finally, summarizing briefly the current POMDP modeling, it consists of 10 actions, 4 observations and 332 states in a discounted infinite horizon format. Based on the same idea of combining maintenance and inspection actions, as just presented, more action choices can be straightforwardly modeled, for example minor-repair-visual-inspection-monitoring-inspection. However, we did not find any reason in doing this at this point.

3. **Perseus algorithm**

As already stated, the Perseus point-based value iteration algorithm, [20], is used in this work in order to compute efficient solutions for this large POMDP formulation. A general description and comments about point-based solvers can be seen in the companion Part I paper. As in every point-based solver, Perseus uses a simple, lower bound approximation in order to initialize the value function over the belief simplex. A commonly used initial value function is given by a single α-vector, [1], with all its components equal to:

$$\alpha(s) = \frac{1}{1-\gamma} \min_{\substack{s \in S \\ \alpha \in A}} R(s,\alpha), \qquad (14)$$

which is guaranteed to be a lower bound. Other initializations are appropriate as well, like:

$$\alpha(s) = \max_{\alpha \in A}\left[\frac{1}{1-\gamma} \min_{s \in S} R(s,\alpha)\right], \qquad (15)$$

or blind policy initialization [37], etc.

Unlike most point-based algorithms, Perseus builds a set of reachable belief points *B* at the beginning and this set remains fixed throughout the complete algorithm. Compared to other algorithms that use various heuristics to collect points, Perseus builds up *B* simply and inexpensively by performing random walks over the belief space, through simulation of random trajectories of possible actions and observations, starting from an initial belief point.

At each Bellman backup step, Perseus starts by sampling a belief point **b** uniformly at random from *B* and computes its updated value and α-vector. Having performed the backup it first checks if the value of **b** is improved. If it is improved it adds this vector to the new value function representation and further checks which other points in *B* have an improved value based on this newly calculated vector. The hope is that the linear vector will improve the value of many other points in *B* and all these points will be removed from $\tilde{B}$, which is an auxiliary set containing the non-improved points. If the value of **b** is not improved the new α-vector is ignored and a copy of the maximizing vector of **b** from V_n is used for V_{n+1} as well. Point **b** is then considered improved and is removed from $\tilde{B}$ together with any other belief point which had the same vector as maximizing one in V_n. This procedure ensures that $\tilde{B}$ shrinks and the backup step will terminate after some iterations, since eventually all the points will be improved and $\tilde{B}$ will become zero. As long as $\tilde{B}$ is not empty the algorithm continues sampling points from it and adding their α-vectors. To summarize the backup stage of the algorithm in pseudocode format we write, [20]:

Perseus backup stage

---

1. Set $|V_{n+1}| = 0$. Initialize $\tilde{B}$ to B.
2. Sample a belief point $\mathbf{b}$ uniformly at random from $\tilde{B}$ and compute its α-vector.
3. If $\mathbf{b} \cdot \alpha \geq V_n(\mathbf{b})$ then add α to V_{n+1}, otherwise add $\alpha' = \arg\max_{\{\alpha_n^i\}_i} \left[\mathbf{b} \cdot \alpha_n^i\right]$ to V_{n+1}.
4. Compute $\tilde{B} = \{\mathbf{b} \in B : V_{n+1}(\mathbf{b}) < V_n(\mathbf{b})\}$.
5. If $\tilde{B} = 0$ stop, else go to 2.

Given that at every backup step the new value function is initialized without any α-vectors and that Perseus only selects vectors that are useful at the current step of the algorithm, no pruning is required. The backup steps continue until some convergence or termination criterion is met. Several criteria can be considered, like a bound on the value function difference between successive estimations, running time of the algorithm, number of α-vectors difference between consecutive value functions, number of belief points that have a different optimal action, etc.

The key idea underlying Perseus is that when a point is backed up the resulting vector improves the value of many other belief points in B, apart from the value of the point that generated it. Based on this observation, the resulting value function has a relatively small number of vectors (in comparison to other solvers) and the backups of Perseus are performed in an asynchronous dynamic programming style since only a small subset of B is randomly visited, at each iteration. By backing up non-improved points Perseus focuses on interesting regions of the reachable belief space and by sampling at random ensures that eventually all points in B will be taken into account. The asynchronous backups further allow the algorithm to use a large B, which has a positive effect on the approximation accuracy as shown in [38]. In Fig. 2 the backup stage of Perseus is illustrated. Having a value function approximation of two vectors and a belief set of 8 points Perseus first selects and backups b_7 and improves the value of all points from b_5 to b_8 in comparison to their previous estimate (dashed lines). Since the value of some points has not improved the algorithm continues by improving b_1 and b_2 and finally finds a new approximate value function (of three vectors now) by backing up b_4 which also improves all the remaining belief points and terminates this backup step. A new backup step then starts and the procedure continues until convergence, as in Fig. 3, where iterative lower bound evaluations of the value function until an approximate, converged value function $\hat{V}^*$ can be seen.

4. **Results**

Having presented the POMDP framework in a detailed manner, simulation results and Perseus performance are now demonstrated and discussed. All numerical experiments were run on a desktop PC with 32-bit windows, Intel Core2Quad processor at 2.33GHz and 3GB RAM.

As stated earlier, Perseus builds a set of reachable belief points *B*, at the beginning of the algorithm, by performing random walks over the belief space, through simulation of random trajectories of possible actions and observations, starting from an initial belief point. For the specific example, a set of 120000 belief points is finally used by performing random simulations, based on action selection probabilities equal to 0.84/3 for all the do-nothing/no-repair actions and 0.08/3 for all minor- and major-repair actions. Every 124 simulation steps each random trajectory was terminated and a new one was initiated from the initial belief point. The initial point of the system is the state with condition level 1 and the first deterioration rate, with certainty. Due to the continuous restarting of new trajectories the replace action was not sampled during the runs. All these parameters were finally chosen because they provide an adequate coverage of the whole reachable belief simplex. For the backup part of the algorithm, we have set Perseus to terminate after it had completed a backup stage beyond 86400s (24h), in order to check the algorithmic behavior for prolonged time.

Fig. 3. Iterative lower bound value function approximations.

Apart from Perseus, the problem is also solved with two methods that are based on the underlying MDP, the Most Likely State (MLS) and the QMDP methods, [1], in order to examine their performance. As mentioned earlier, MDPs form the current computational framework used by state agencies for asset management. To evaluate the computed policies from all methods 100 different trajectories of 500 steps each, based on the policy this time, have been simulated and their averaged expected reward is calculated. Unless differently stated all the simulations started from the state with no deterioration (condition level 1 and the first deterioration rate).

In Fig. 4 all the supported actions by the 2319 α-vectors, that compose the final value function of the algorithm, are shown. As seen, the calculated optimum policy includes all the actions except the major-repair-monitoring-inspection action since Perseus did not identify any area of the belief simplex where this action would have been the optimal one for the decision-maker to take. One simple explanation for this can be derived after observing the two, randomly selected, example simulations of the computed policy, in Figs. 5-6. In these two figures it is seen that based on the policy the decision-maker chooses to perform major-repair actions in relatively early deterioration stages. Therefore, since the structure will be in relatively good condition, especially after maintenance, expensive inspection techniques, following an already expensive maintenance action, are not justified. The fact that the decision-maker chooses major-repair actions in relatively early deterioration stages may seem surprising at first. The reason however is that major-repair actions are considerably more expensive than minor-repair ones but they have the advantage of both improving the deterioration rate and the condition level of the structure. Due to this, the decision-maker executes these actions early enough, in order to take advantage of both the low deterioration rates and the deterioration rate improvement feature of the actions, and to guide the structure to an even lower deterioration rate, eventually prolonging its life and minimizing the life-cycle cost. Yet, in later, increased deterioration rates, major repair actions are not usually preferred because the deterioration rate will still improve by these actions, but since the structure is already in high rates this small improvement does not justify such expensive maintenance and thus minor-repair actions are been chosen instead. This planning pattern of the decision-maker is of course dependent on the different capabilities of the actions to improve the system and their relative, to each other, costs. Nevertheless, it is very interesting to observe this pattern since it is rather different than what it is often done currently in practice, where decision-making is much more myopic and greedier strategies are followed (defined in an optimization context).

Other interesting conclusions can be also inferred from Figs. 5 and 6. First of all, as seen, the current modeling can also indicate the best time to renew the structure. Although in these two, randomly selected, simulations the renewal time is around 150 policy steps, we have seen others, where based on observations the structure should be replaced as early as 55 steps or as late as 300 steps. Furthermore, this modeling feature can be also used, in a different context, as a very useful prognostic tool towards estimations of the most profitable structural service period. Another appealing observation is that at increased deteriorated conditions the computed policy indicates that generally the structure should be inspected every one to three years, depending on the observations each time, when was the last maintenance action performed, etc. Currently, many structural assets (e.g. bridge decks) are periodically inspected biannually. The current study shows that for such non-flexible policies (periodic inspections are imposed on the management policy) this 2 year interval might be a good choice. However, this planning is certainly sub-optimal (in comparison to the approach in this paper), both in safety and economic terms, because there will certainly be cases where more or less frequent inspection visits are necessary. Apart from suggesting inspection visits, the computed policy also manages successfully, in each step, the required

inspection quality. Generally, in cases when there is high uncertainty about the condition of the structure the decision-maker will choose the more expensive and accurate monitoring-inspection than the less expensive and accurate visual one. Other cases where monitoring-inspection is usually preferred is when the decision-maker wants to have increased certainty about the condition of the structure, before an investment on an expensive maintenance action (major-maintenance) is decided to be made or not. This conclusion is not frequently met before a renewal action however, since the structure is in a poor state and a visual inspection usually suffices to determine its replacement. Overall, the versatility of the policy in order to achieve optimality should be emphasized at this point. Without imposing any unjustified constraints on the policy search space (like periodic inspection periods, threshold performances, perfect inspections and many more) in order to make the optimization method capable of solving the problem, the proposed approach finds a complex policy strategy. This strategy is successful because it is based on the most important information the decision-maker may have at each step, which is the belief over the conditions of the structure.

Perseus results are also compared with results from the MLS and QMDP methods. As these two methods are based on the underlying MDP they never include informative gathering actions in their computed policies. It is easily understood that this is already a big limitation of the methods in this application, since the problem is cast so that the decision-maker has to choose when and how to inspect the structure. In Figs. 7-8 the averaged expected reward of the MLS and QMDP policies, calculated as mentioned earlier, can be seen. Results in Fig. 7 are based on simulations that started at the usual initial point of no deterioration. For the results in Fig. 8, simulations started at an initial belief point of 51% and 49% probability of being in a condition level 2 and 3, respectively, at a deterioration rate of 68. Looking at Figs. 7-8 it is easily concluded that the policies computed by the two methods are approximately equal, in all cases. The no-inspection bars represent results based on the already presented modeling. These results are named no-inspection because the methods never choose actions that include inspections in their policies. To investigate the quality of the policies in more realistic situations, where inspections are obviously performed, the problem is reformulated so as the decision-maker is obliged to inspect in every step. Given that MDPs cannot also choose actions based on how good their inspection quality is, the problem is reformulated twice, allowing either only uncertain visual-inspection actions or uncertain monitoring-inspection actions. An interesting outcome from all these different cases is that the no-inspection modeling produces better policies if the structure starts from intact conditions, taking advantage of the small uncertainties in the beginning of this case and the discount factor. This result is not so obvious anymore when the initial conditions of the structure are in a deteriorated, uncertain state. In other words, when the uncertainty and the deterioration are low, inspections are not needed that much and decision mistakes due to ignorance about the true condition of the structure are counterbalanced by the savings of not paying for inspections. However, in deteriorated, uncertain situations, inspections are useful and this is why in Fig. 8 the simulations with inspections give an equal reward to the no-inspection case, although the decision-maker pays annually for inspections in these cases. These findings clearly indicate that the MLS and QMDP methods are not able to find a policy that will be optimum for the whole service life of the structure in this problem, mainly due to the known MDP limitations. Inspections are not needed that much in the beginning of the structure's service period but are desperately needed when deterioration and uncertainty are present. As stated in section 4.1 in [1], we could have further reformulated the problem, seeking better solutions with these methods, by dropping the requirement for annual visits to the structure. This would allow greater flexibility, but even so these methods still would not have been able to choose different inspection techniques or to separate maintenance from inspection actions (and vice versa) in any decision epoch.

To compare the policy performance of these methods with Perseus results, which take advantage of the full POMDP modeling, the no-inspection case is chosen, starting from intact initial conditions, since this is the case that gives the highest rewards. In Fig. 9 comparison results can be seen and the superiority of Perseus (and of the POMDP modeling) over the two methods (and the MDP modeling) is obvious.

Perseus performance at each backup stage can be seen in Figs. 10, 11 and 12. In Fig. 10 the expected discounted reward is shown, based on the computed policy in each step and averaged over the 100 performed simulations, and in Fig. 11 the averaged expected total reward is seen. Both figures show an overview of Perseus performance, as well as a detailed view of the last part of the iteration stages. The QMDP solution (no inspection case) is also shown in both figures. QMDP calculates a solution very quickly but as

already explained it is not the optimal one. Perseus takes around 4minutes in the discounted case to permanently exceed the performance of QMDP and around 18seconds in the total reward case. In this latter case, where future rewards are not discounted, QMDP shows its weaknesses earlier, as explained before. Relying on the simulation results, it seems that Perseus reached an adequate policy in around 30minutes and it finally converges satisfactorily after approximately 4hours of computational time. The overall performance of Perseus is typical of point-based POMDP solvers that they alter their policy considerably during the backup stages, while updating and improving the value function, until they reach some sort of convergence. In Fig. 12 the number of hyperplanes that compose the value function in each step can be seen. This number has an increasing trend, as more backup stages are performed, and reached a value of 2319 in the last stage. The fact that Perseus keeps adding α-vectors to its value function representation is an indication that it gradually improves it. It might be possible that a better policy and higher rewards are obtained after some more backup stages and computational time. This is something typical of point-based POMDP solvers as well, i.e. the fact that may seem to have reached a plateau for some time and then they suddenly improve the policy and a jump in the rewards can be noticed. In order to know if that is the case in this application however, an upper-bound value function representation should have been computed as well, so as to check where the current solution lies between the bounds. Nonetheless, only the lower bound value is computed in this work and shown in Fig. 12. In the overall view, the average bound value is illustrated, considering all the points collected by Perseus. That means that the bound for each point is calculated, summed up and divided by the total number of points. As seen, the algorithm has been initiated with a very low initial value function representation, to guarantee its lower bound attribute, and gradually Perseus updates and improves the bound. Finally, by comparing the simulation results with the lower bound value of the initial point it is seen that after the value function reaches a satisfactorily high level and seems converged, simulation results are consistent with the bound and normally, barely exceed it.

5. Conclusions

The problem of planning an optimum life-cycle cost policy that can suggest inspection/monitoring and maintenance actions based on the structural conditions in real time is considered in this work, complementing the Part I companion paper. The optimum life-cycle policies are determined based on Partially Observable Markov Decision processes (POMDPs), involving stochastic control, probabilistic models, uncertain structural data and Bayesian principles. A detailed application example for a corroding reinforced concrete structure is analyzed and an advanced infinite horizon formulation is cast and solved, with non-stationary stochastic phenomena, connection to physically based stochastic models and a considerably large state-space of 332 states. Choice availability of different monitoring and maintenance actions, uncertain observation and action outcomes, non-periodic structural visits and the cost-benefit of information are also incorporated in the formulation, resulting in a highly complex optimum policy, which cannot be achieved by any other method, and combines a range of inspection/monitoring types and intervals, and maintenance actions and action times. To be able to solve this challenging problem we resort to point-based solvers and particularly asynchronous dynamic programming and Perseus. Apart from Perseus we also solve the problem with simple approximate solvers, based on MDPs, demonstrating differences in performance and quality of solutions and touching upon the important notion of the value of information. Synoptically, the current paper provides a generic framework that modernizes the way relevant problems are solved today, sets a step forward in large-scale modeling, showcases deployment of point-based POMDP methods and motivates their use in a wider variety of problems and practical applications.

The biggest limitation of the presented approach however is the substantial computational demands for large state-spaces both in terms of computational memory and time. As such, research for dedicated models and solvers that can efficiently scale up the representation of the problem even more should be continued. Future research work is also possible in several other directions, as for example a more rigorous, systematic formation of structural maintenance actions in a POMDP context.

Acknowledgements

The work reported in this paper has been partially supported by the National Science Foundation under Grant No. CMMI-1233714. This support is gratefully acknowledged.

References

[1] Papakonstantinou KG, Shinozuka M. Planning structural inspection and maintenance policies via dynamic programming and Markov processes. Part I: Theory. Reliability Engineering and System Safety. *Under Review*.
[2] Frangopol D, Kallen M-J, Noortwijk J. Probabilistic models for life-cycle performance of deteriorating structures: review and future directions. Progress in Structural Engineering and Materials. 2004;6.
[3] Golabi K, Kulkarni R, Way G. A statewide pavement management system. Interfaces. 1982;12.
[4] Hawk H, Small E. The BRIDGIT bridge management system. Structural Engineering International. 1998;8.
[5] Adey B, Klatter L, Kong J. Overview of existing bridge management systems. The IABMAS Bridge Management Committee. 2010.
[6] Golabi K, Thompson P, Hyman W. Pontis technical manual prepared for Federal Highway Administration. 1992.
[7] Golabi K, Shepard R. Pontis: A system for maintenance optimization and improvement of US bridge networks. Interfaces. 1997;27.
[8] Thompson P, Small E, Johnson M, Marshall A. The Pontis bridge management system. Structural Engineering International. 1998;8.
[9] Madanat S, Ben-Akiva M. Optimal inspection and repair policies for infrastructure facilities. Transportation Science. 1994;28.
[10] Smilowitz K, Madanat S. Optimal inspection and maintenance policies for infrastructure networks. Computer-Aided Civil and Infrastructure Engineering. 2000;15.
[11] Ellis H, Jiang M, Corotis R. Inspection, maintenance, and repair with partial observability. Journal of Infrastructure Systems, ASCE. 1995;1.
[12] Jiang M, Corotis R, Ellis H. Optimal life-cycle costing with partial observability. Journal of Infrastructure Systems, ASCE. 2000;6.
[13] Faddoul R, Raphael W, Chateauneuf A. A generalised partially observable Markov decision process updated by decision trees for maintenance optimisation. Structure and Infrastructure Engineering. 2011;7.
[14] Byon E, Ntaimo L, Ding Y. Optimal maintenance strategies for wind turbine systems under stochastic weather conditions. IEEE Transactions on Reliability. 2010;59.
[15] Zouch M, Yeung TG, Castanier B. Optimal resurfacing decisions for road maintenance: A POMDP perspective. Reliability and Maintainability Symposium (RAMS)2011.
[16] Srinivasan R, Parlikad AK. Value of condition monitoring in infrastructure maintenance. Computers & Industrial Engineering. 2013;66.
[17] AlDurgam MM, Duffuaa SO. Optimal joint maintenance and operation policies to maximise overall systems effectiveness. International Journal of Production Research. 2013;51.

[18] Robelin C-A, Madanat S. History-dependent bridge deck maintenance and replacement optimization with Markov decision processes. Journal of Infrastructure Systems, ASCE. 2007;13.
[19] Corotis R, Ellis H, Jiang M. Modeling of risk-based inspection, maintenance and life-cycle cost with partially observable Markov decision processes. Structure and Infrastructure Engineering. 2005;1.
[20] Spaan M, Vlassis N. Perseus: Randomized point-based value iteration for POMDPs. Journal of Artificial Intelligence Research. 2005;24.
[21] Papakonstantinou KG, Shinozuka M. Optimum inspection and maintenance policies for corroded structures using Partially Observable Markov Decision Processes (POMDPs) and stochastic, physically based models. *Under Review*.
[22] Andrade C, Alonso C. Test methods for on-site corrosion rate measurement of steel reinforcement in concrete by means of the polarization resistance method. Materials and Structures. 2004;37.
[23] Song H, Saraswathy V. Corrosion monitoring of reinforced concrete structures - A review. International Journal of Electrochemical Science. 2007;2.
[24] Razaqpur A, Isgor O. Prediction of reinforcement corrosion in concrete structures. In: Chen S-S, Ang AH-S, editors. Frontier technologies for infrastructure engineering, Structures and infrastructures series Vol4: CRC Press; 2009.
[25] Papakonstantinou KG, Shinozuka M. Probabilistic model for steel corrosion in reinforced concrete structures of large dimensions considering crack effects. Engineering Structures. *In Print*.
[26] Madanat S, Mishalani R, Ibrahim W. Estimation of infrastructure transition probabilities from condition rating data. Journal of Infrastucture Systems, ASCE. 1995;12.
[27] DeStefano P, Grivas D. Method for estimating transition probability in bridge deterioration models. Journal of Infrastructure Systems, ASCE. 1998;4.
[28] Mauch M, Madanat S. Semiparametric hazard rate models of reinforced concrete bridge deck deterioration. Journal of Infrastructure Systems, ASCE. 2001;7.
[29] Mishalani R, Madanat S. Computation of infrastructure transition probabilities using stochastic duration models. Journal of Infrastructure Systems, ASCE. 2002;8.
[30] Guignier F, Madanat S. Optimization of infrastructure systems maintenance and improvement policies. Journal of Infrastructure Systems, ASCE. 1999;5.
[31] AASHTO. Guide for commonly recognized structural elements. American Association of State Highway and Transportation Officials; 2002.
[32] Broomfield J. Corrosion of steel in concrete. New York: Taylor & Francis; 2007.
[33] Song H, Shim H, Petcherdchoo A, Park S. Service life prediction of repaired concrete structures under chloride environment using finite difference method. Cement & Concrete Composites. 2009;31.
[34] Caltrans. Contract cost data. State of California Department of Transportation. 2006.
[35] Thompson P, Soares R, Choung H, Najafi F, Kerr R. User cost model for bridge management systems. Transportation Research Record. 2000;1697.
[36] Woodward R. Bridge management in Europe. European Commission, Project Final Report. 2001.

[37] Hauskrecht M. Value-function approximations for partially observable Markov decision processes. Journal of Artificial Intelligence Research. 2000;13.
[38] Pineau J, Gordon G, Thrun S. Point-based value iteration: An anytime algorithm for POMDPs. International Joint Conference on Artificial Intelligence (IJCAI)2003.

- An advanced, non-stationary, 332 state, infinite horizon POMDP formulation is solved.
- The cost-benefit of information is naturally incorporated in the methodology.
- The formation and solution of the problem modernize and extend relevant approaches.
- The estimated complex optimum policy cannot be achieved by any other method.
- The suggested framework is compared with simpler techniques based on MDPs (MLS, QMPD)

Fig. 1. Probability of being in any condition state, starting from sound condition, and comparison of physically based stochastic modeling with Markov chain simulations.

Fig. 2. Perseus backup stage.

Fig. 4. Actions supported by the 2319 hyperplanes.

Fig. 5. First example simulation.

Fig. 6. Second example simulation.

Fig. 7. Comparison of policies. The simulations initialized in the no deterioration state.

Fig. 8. Comparison of policies. The simulations initialized in states with deteriorated conditions.

Fig. 9. Comparison of policies and superiority of POMDP.

Fig. 10. Perseus performance. Average expected discounted reward.

Fig. 11. Perseus performance. Average expected undiscounted reward.

Fig. 12. Perseus performance. Number of hyperplanes and lower bounds.

Fig01

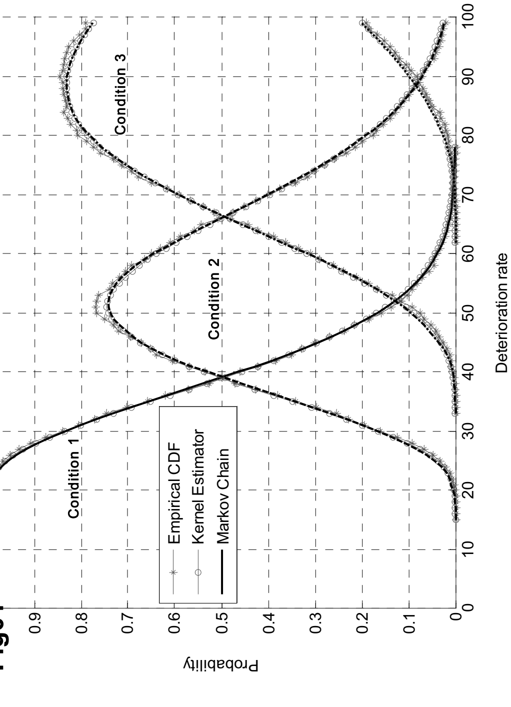
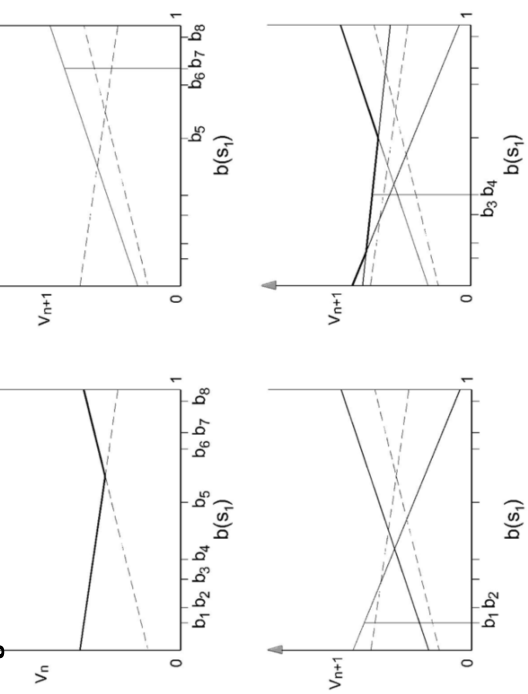

Probability
Deterioration rate
Empirical CDF
Kernel Estimator
Markov Chain
Condition 1
Condition 2
Condition 3
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
10
20
30
40
50
60
70
80
90
100

Fig02



Vn
Vn+1
Vn+1
Vn+1
b(s1)
b(s1)
b(s1)
b(s1)
b1 b2 b3 b4 b5 b6 b7 b8
b5 b6 b7 b8
b1 b2
b3 b4
0
1

Fig03

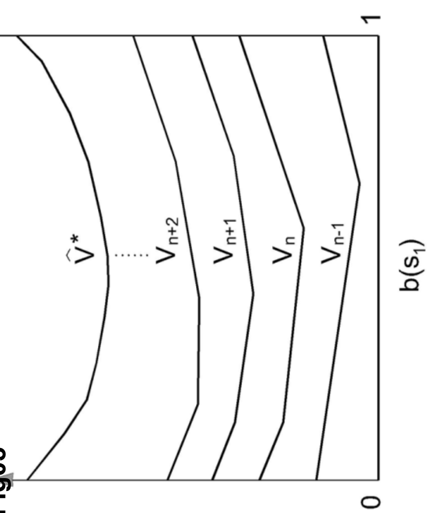


Fig04

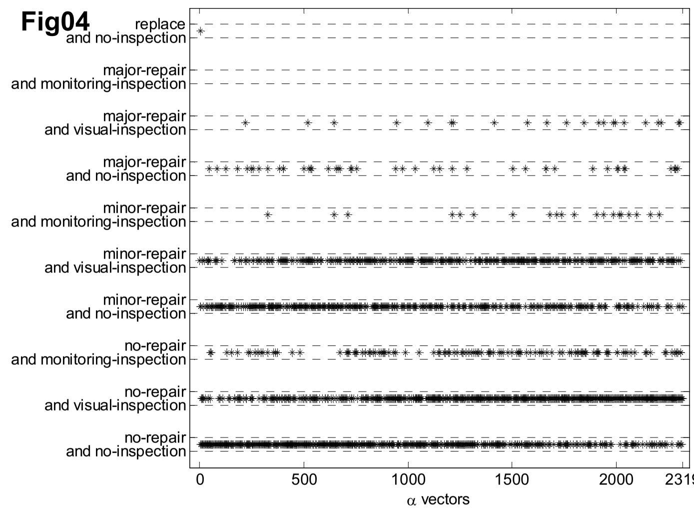

replace
and no-inspection
major-repair
and monitoring-inspection
major-repair
and visual-inspection
major-repair
and no-inspection
minor-repair
and monitoring-inspection
minor-repair
and visual-inspection
minor-repair
and no-inspection
no-repair
and monitoring-inspection
no-repair
and visual-inspection
no-repair
and no-inspection
0
500
1000
1500
2000
2319
α vectors

Fig05

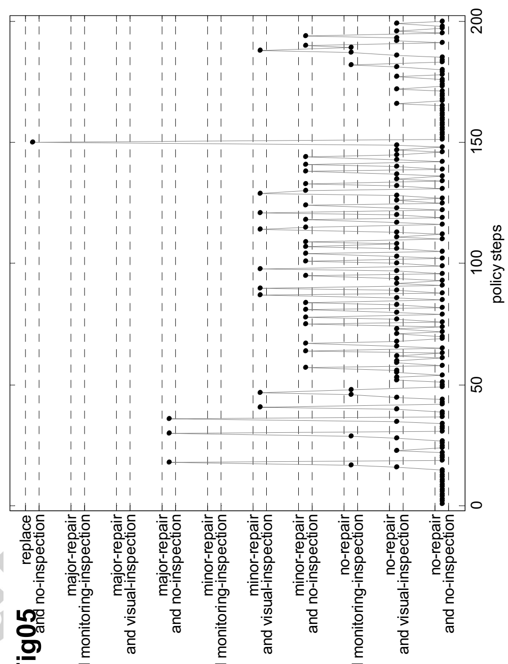

replace
and no-inspection
major-repair
and monitoring-inspection
major-repair
and visual-inspection
major-repair
and no-inspection
minor-repair
and monitoring-inspection
minor-repair
and visual-inspection
minor-repair
and no-inspection
no-repair
and monitoring-inspection
no-repair
and visual-inspection
no-repair
and no-inspection
0
50
100
150
200
policy steps

Fig06

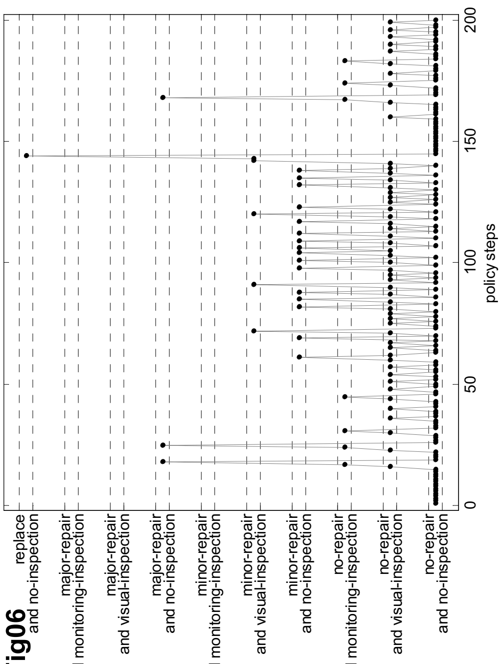

replace
and no-inspection
major-repair
and monitoring-inspection
major-repair
and visual-inspection
major-repair
and no-inspection
minor-repair
and monitoring-inspection
minor-repair
and visual-inspection
minor-repair
and no-inspection
no-repair
and monitoring-inspection
no-repair
and visual-inspection
no-repair
and no-inspection
0
50
100
150
200
policy steps

Fig07

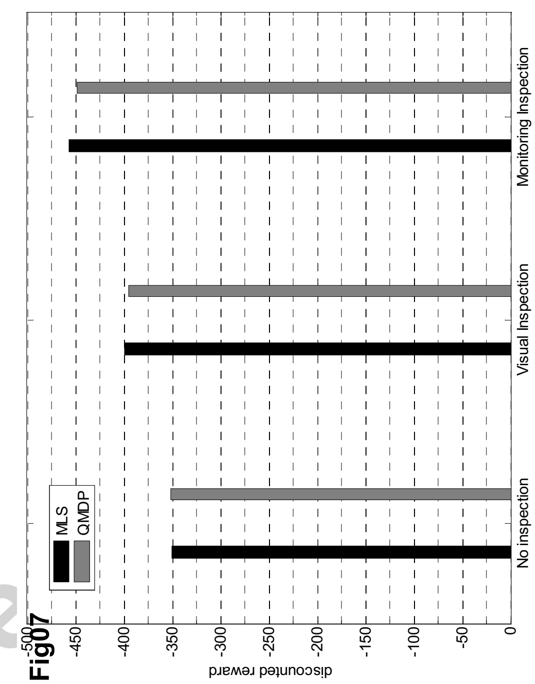
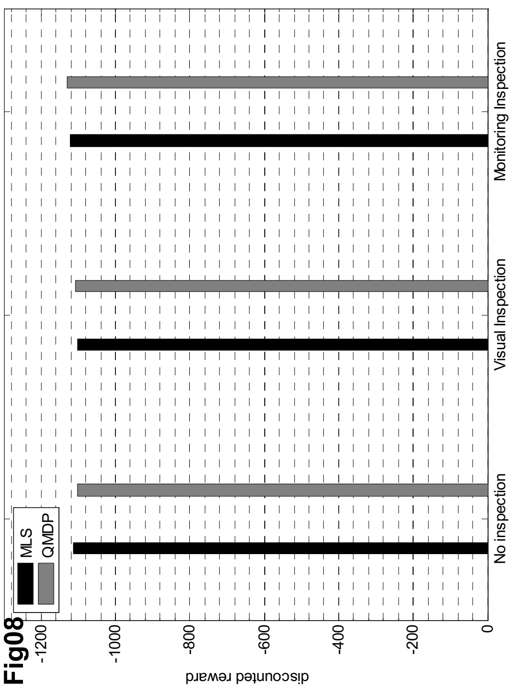
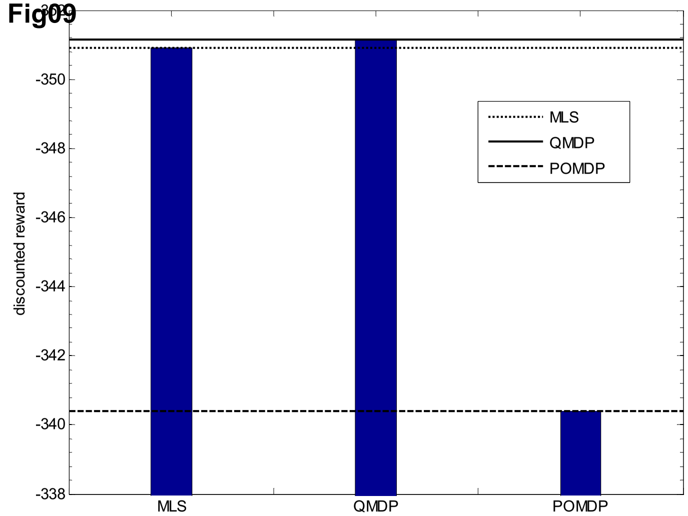

MLS
QMDP
discounted reward
-500
-450
-400
-350
-300
-250
-200
-150
-100
-50
0
No inspection
Visual Inspection
Monitoring Inspection

Fig08



discounted reward
-1200
-1000
-800
-600
-400
-200
0
MLS
QMDP
No inspection
Visual Inspection
Monitoring Inspection

Fig09



discounted reward
-350
-348
-346
-344
-342
-340
-338
MLS
QMDP
POMDP
MLS
QMDP
POMDP

Fig10

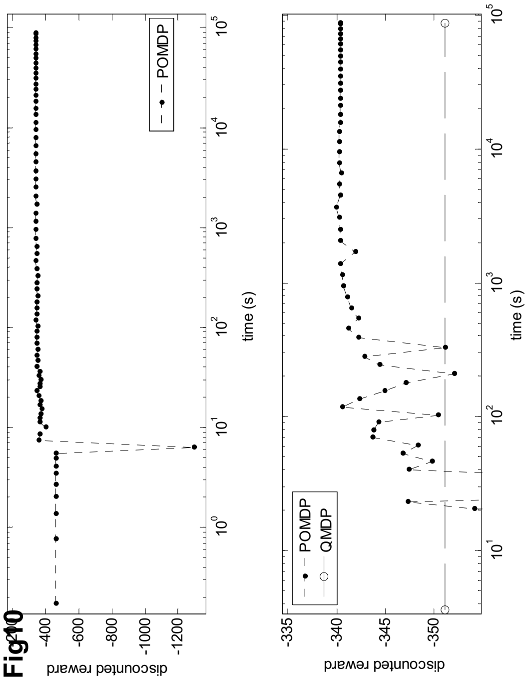


Fig11

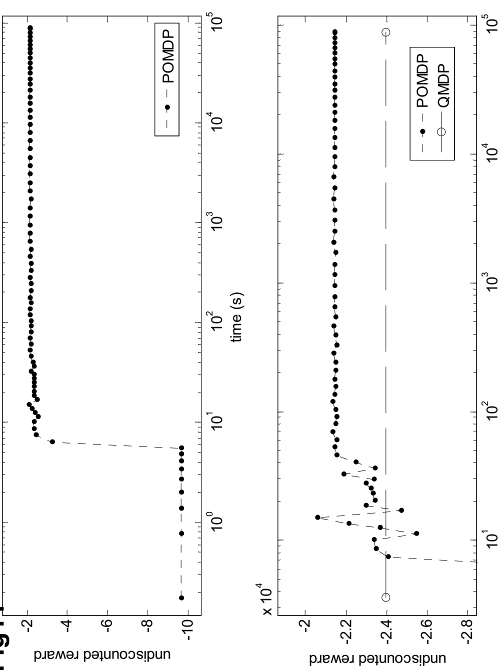
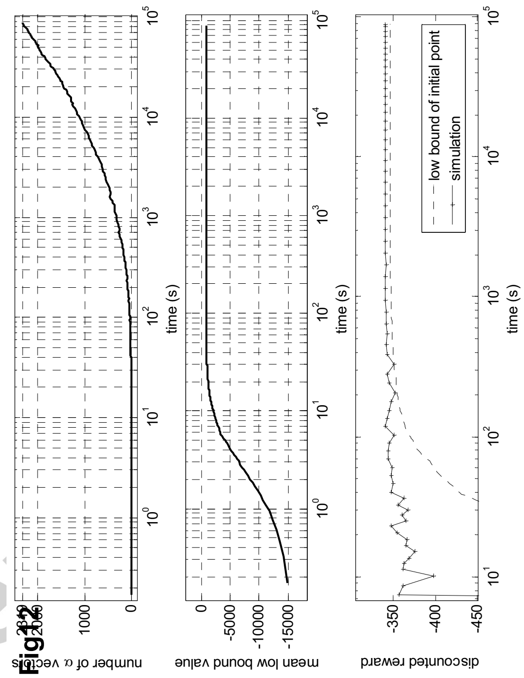


Fig12


number of α vectors
mean low bound value
discounted reward
time (s)
low bound of initial point
simulation

Maintenance actions rewards

Condition levels	1	2	3	4
Do-nothing	0.00	0.00	0.00	0.00
Minor-repair	-60.00	-110.00	-160.00	-280.00
Major-repair	-105.00	-195.00	-290.00	-390.00
Replace	-820.00	-820.00	-820.00	-820.00

Inspection actions rewards

Condition levels	1	2	3	4
No-inspection	0.00	0.00	0.00	0.00
Visual-inspection	-4.50	-4.50	-4.50	-4.50
Monitoring-inspection	-7.50	-7.50	-7.50	-7.50

User / Penalty rewards

Condition levels			
1	2	3	4
-5.00	-40.00	-120.00	-250.00