A Predictive Maintenance System for Integral Type Faults based on Support Vector Machines: an Application to Ion Implantation

**Gian Antonio Susto** (i-ii), Andrea Schirru (ii), Simone Pampuri (i-ii), Daniele Pagano (iii), Seán McLoone (i-iv), Alessandro Beghi (v)

(i) National University of Ireland, Maynooth (NUIM) - (ii) SpeedUp Consulting, Italy - (iii) STMicroelectronics, Catania, Italy - (iv) Queen’s University of Belfast, UK - (v) University of Padova, Italy

CASE 2013 Madison - Aug. 18th, 2013
Maintenance Policies

- Maintenance Management
  - Run-to-Failure
  - Preventive
  - Predictive

EFFICIENCY

SIMPLICITY
1) Run-to-Failure Maintenance - R2F

When repairs or restore actions are performed only after the occurrence of a failure

“If it’s not broken don’t fix it”

Common Policy in the fabs
2) Preventive Maintenance

Maintenances carried out on a planned schedule with the aim of anticipating the process failures.

Failures are usually warded off but unnecessary maintenances are performed.
3) Predictive Maintenance - PdM

Maintenance actions are taken after the verification of conditions indicating the degradation of the process/equipment. A PdM system predicts when such actions have to be taken.

Proposed Policy
Techniques for Predictive Maintenance (PdM)

- No general approaches for PdM, tailored to specific problems
- Several techniques may be suitable for PdM i.e.
  1. [Wu 2007]: regression methods (Neural Networks, Elastic Nets)
  2. [Baly 2012]: classification methods (Support Vector Machines)
  3. [Pampuri 2011]: survival models
  4. [Butler 2010]: filtering and prediction (Particle Filters)

S. Wu, N. Gebraeel, M. Lawley, Y. Yih
A Neural Network Integrated Decision Support System for Condition-Based Optimal Predictive Maintenance Policy

R. Baly, H. Hajj
Wafer Classification Using Support Vector Machines

S. Pampuri, A. Schirru, C. De Luca, G. De Nicolao
Proportional Hazard Model with L1 Penalization Applied to Predictive Maintenance in Semiconductor Manufacturing

S. Butler, J. Ringwood
Particle Filters for Remaining Useful Life Estimation of Abatement Equipment Used in Semiconductor Manufacturing
*IEEE Conference on Fault-Tolerant Systems*, Nice, 6-8 Oct, 2010
Integral Type Faults and Problem at Hand

Integral Type Faults

Faults caused by machine usage
(i.e. stress on mechanical tool parts, chambers progressively becoming dirty, etc.)

Problem at hand: breakings of tungsten filament in ion-implanters
Desired PdM Module

- **Goal**: given the availability (free) of process variables $X \in \mathbb{R}^{1 \times p}$ (pressures, temperatures, currents, etc.) for each process run - define an indicator $y$ of the current state of the maintenance issue (**health factor**)

$$y = f(X)$$

- Treated as a **classification problem**: 2 qualitative classes
  
  (i) **faulty** wafers
  
  (ii) **non-faulty** wafers

- **Support Vector Machines (SVMs)**: estimate to which class a new observation (wafer) belongs, providing a decision boundary
Desired PdM Module

**Goal**: given the availability (free) of process variables $X \in \mathbb{R}^{1 \times p}$ (pressures, temperatures, currents, etc.) for each process run - define an indicator $y$ of the current state of the maintenance issue (**health factor**)

$$y = f(X)$$

- Treated as a **classification problem**: 2 qualitative classes
  1. **faulty** wafers
  2. **non-faulty** wafers
- **Support Vector Machines (SVMs)**: estimate to which class a new observation (wafer) belongs, providing a **decision boundary**
Algorithm Concept

- **Algorithm Main Concept**: exploit the distance of a new wafer from the SVM-defined decision boundary as an indicator of ‘distance’ to the fail

- **Insight/Main Assumption**:
  1. if an observation/wafer is well inside the non-faulty class then it is far from being faulty
  2. if a wafer is close to the decision border then it is close to a faulty condition
**Algorithm Concept**

- **Algorithm Main Concept**: exploit the distance of a new wafer from the SVM-defined decision boundary as an indicator of ‘distance’ to the fail

- **Insight/Main Assumption**:
  1. If an observation/wafer is well inside the non-faulty class then it is far from being faulty
  2. If a wafer is close to the decision border then it is close to a faulty condition
Algorithm Concept

- **Algorithm Main Concept**: exploit the distance of a new wafer from the SVM-defined decision boundary as an indicator of ‘distance’ to the fail

- **Insight/Main Assumption**:
  1. if an observation/wafer is well inside the non-faulty class then it is far from being faulty
  2. if a wafer is close to the decision border then it is close to a faulty condition
Two linearly separable classes to be classified ($A$ and $B$); a Training Set $S$:

$$S : \left\{ x_i \in \mathbb{R}^{1 \times p}, y_i \in \{-1, 1\} \right\}_{i=1}^{n}, \text{ where } y_i = \begin{cases} 1 & i\text{-th sample } \in A \\ -1 & i\text{-th sample } \in B \end{cases}$$

We define the hyperplane $F_0 \in \mathbb{R}^p$:

$$F_0 = \{ x | f(x) = x\beta + \beta_0 = 0 \}$$

Classification based on $f(x)$ ($F_0$): for a new sample $x^{\text{new}} \notin S$

$$y^{\text{new}} = \begin{cases} \text{Class A (1) if } f(x^{\text{new}}) > 0 \\ \text{Class A (-1) if } f(x^{\text{new}}) < 0 \end{cases}$$

How to compute an 'optimal' $f(x)$?
We choose the hyperplane that gives the biggest margin between classes:

\[
\begin{align*}
    \max_{\beta, \beta_0, \|\beta\|} & \quad M, \\
    \text{subject to} & \quad y_if(x) \geq M, \ i = 1, \ldots, n
\end{align*}
\]

\[
\begin{align*}
    \max_{\beta, \beta_0} & \quad \beta, \beta_0 \\
    \text{subject to} & \quad y_if(x) \geq 1, \ i = 1, \ldots, n
\end{align*}
\]

(convex optimization problem)

Related Lagrange function:

\[
L = \frac{1}{2} \|\beta\|^2 - \sum_{i=1}^{n} \alpha_i [y_i (x_i \beta + \beta_0) - 1].
\]

Wolfe Dual form (deriving w.r.t. \(\beta, \beta_0\), setting the derivatives to zero, substituting back)

\[
L_D = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j x_i x_j^T
\]

subject to \(\alpha_i \geq 0, \ i = 1, \ldots, n\)

Solution maximizing \(L_D\) in the positive orthant
The event that triggers the maintenance event is

\[ f(x) < \tau \in \mathbb{R}^+ \]
PdM: Performance Evaluation

- The event that triggers the maintenance event is
  \[ f(x) < \tau \in \mathbb{R}^+ \]

- The performance of a PdM policy should be evaluated in terms of:
  
  (i) # Unexpected Breaks \( N_{UB} \)
  
  (ii) # Unexploited Lifetime \( N_{UL} \)
The event that triggers the maintenance event is

\[ f(x) < \tau \in \mathbb{R}^+ \]

The performance of a PdM policy should be evaluated in terms of:

(i) # Unexpected Breaks \( N_{UB} \)
(ii) # Unexploited Lifetime \( N_{UL} \)

Different associated costs (generally \( C_{UB} \gg C_{UL} \)), the total cost should be minimized

\[ J(\tau, t) = N_{UB}(\tau) \times C_{UB}(t) + N_{UL}(\tau) \times C_{UL}(t) \]
The event that triggers the maintenance event is

\[ f(x) < \tau \in \mathbb{R}^+ \]

The performance of a PdM policy should be evaluated in terms of:

(i) # Unexpected Breaks \( N_{UB} \)
(ii) # Unexploited Lifetime \( N_{UL} \)

Different associated costs (generally \( C_{UB} \gg C_{UL} \)), the total cost should be minimized

\[ J(\tau, t) = N_{UB}(\tau) \times C_{UB}(t) + N_{UL}(\tau) \times C_{UL}(t) \]

Dependence of time for the costs and on \( \tau \) for the performances
The event that triggers the maintenance event is

\[ f(x) < \tau \in \mathbb{R}^+ \]

The performance of a PdM policy should be evaluated in terms of:

(i) # Unexpected Breaks \( N_{UB} \)
(ii) # Unexploited Lifetime \( N_{UL} \)

Different associated costs (generally \( C_{UB} \gg C_{UL} \)), the total cost should be minimized

\[ J(\tau, t) = N_{UB}(\tau) \times C_{UB}(t) + N_{UL}(\tau) \times C_{UL}(t) \]

Dependence of time for the costs and on \( \tau \) for the performances

PvM and PdM approaches have different performances depending on threshold choice, while R2F has:

(i) \( N_{UB} = 1 \)
(ii) \( N_{UL} = 0 \)
Cross-Validation and PdM

- Costs may change over time, performances $N_{UB}$ and $N_{UL}$ should be provided to dynamically change the maintenance policy
- $N_{UB}$ and $N_{UL}$ evaluated thorough Cross-Validation

```
Algorithm 1: SVM based-PdM module

Data: $X, Y, C_{UB}, C_{UL}, K, q$
Result: Maintenance Rule (defined by $\tau^*$), $N_{UB}, N_{UL}$, $J(T)$

1. Compute and Tune SVM $f(\cdot)$ through Cross-Validation
2. Define a set of threshold values $T \in \mathbb{R}^d$
3. Let $\tilde{N}_{UB} = [\cdot]$ and $\tilde{N}_{UL} = [\cdot]$ (empty vectors)
   for $j = 1$ to $K$ do
     4. Randomly split the data between training and validation, keeping the ratio $q$
     5. Compute $N_{UB}$ and $N_{UL}$ for all entries in $T$
     6. $\tilde{N}_{UB} = [\tilde{N}_{UB}; N_{UB}]$ and $\tilde{N}_{UL} = [\tilde{N}_{UL}; N_{UL}]$
    7. $N_{UB} = \text{Mean}(\tilde{N}_{UB})$ and $N_{UL} = \text{Mean}(\tilde{N}_{UL})$
   8. Compute $J(\tau)$ for all entries in $T$: $J(T)$
   9. Let $\tau^* = \min_{\tau \in T} J(\tau)$

Monte Carlo cross-validation: $K$ simulations with random splitting of the data between

- (i) training [$Nq$ data]
- (ii) test [$N(1 - q)$ data]

- Performances provided as mean over the $K$ simulations

In our tests

\[ K = 1000 \]
\[ q = 0.7 \]
Experimental Settings

- We have available
  
  (i) \( N = 33 \) maintenance cycles: tool period from one maintenance to another with R2F policy
  
  (ii) \( n = 3671 \) run
  
  (iii) \( p = 125 \) physical variables

- PdM compared with PvM based on
  
  (i) the mean \( \mu \) of the maintenance cycle length
  
  (ii) the median \( \eta \) of the maintenance cycle length

- PdM based on two types of SVM
  
  (i) Linear
  
  (ii) Radial-Basis Function (RBF)

- A second level of crossvalidation to tune the SVMs parameters by minimizing the Missclassification Error
Experimental Results

- Averaged performances for various $\tau$
- PdM outperform PvM on both indicators
- RBF guarantees better performances than linear at the cost of a more time consuming tuning

We compute the index (with best threshold $\tau$)

$$J(C_{UB}, C_{UL}) = \min_{\tau} J(\tau, C_{UB}, C_{UL})$$

$J(PdM) - J(PvM)$ is almost always negative

$\Rightarrow$ PdM outperforms PvM
Conclusions

- A PdM for Integral Type faults has been presented.
Conclusions

- A PdM for Integral Type faults has been presented
- The algorithm exploits the availability of process variables and R2F data
Conclusions

- A PdM for Integral Type faults has been presented
- The algorithm exploits the availability of process variables and R2F data
- The PdM system is based on classification methods (SVMs: linear and RBF)
Conclusions

- A PdM for Integral Type faults has been presented
- The algorithm exploits the availability of process variables and R2F data
- The PdM system is based on classification methods (SVMs: linear and RBF)
- The Algorithm provides an assessment of the trade-off between unexploited lifetime and unexpected breaks
Conclusions

- A PdM for Integral Type faults has been presented
- The algorithm exploits the availability of process variables and R2F data
- The PdM system is based on classification methods (SVMs: linear and RBF)
- The Algorithm provides an assessment of the trade-off between unexploited lifetime and unexpected breaks

- The user can dynamically change the maintenance trigger based on the current cost
A PdM for Integral Type faults has been presented

The algorithm exploits the availability of process variables and R2F data

The PdM system is based on classification methods (SVMs: linear and RBF)

The Algorithm provides an assessment of the trade-off between unexploited lifetime and unexpected breaks

The user can dynamically change the maintenance trigger based on the current cost

Tested on a semiconductor manufacturing dataset
Thank you for your attention!

A Predictive Maintenance System for Integral Type Faults based on Support Vector Machines: an Application to Ion Implantation

Gian Antonio Susto, Andrea Schirru, Simone Pampuri, Daniele Pagano, Seán McLoone and Alessandro Beghi

ACKNOWLEDGMENT - The financial support of the Irish Centre for Manufacturing Research and Enterprise Ireland are gratefully acknowledged