Implementing the Matrix Exponential Function on Embedded Processors

James Lebak (presenter) and Andrea Wadell
MIT Lincoln Laboratory*

April 22, 2004

The solution to a differential equation of the form

$$\dot{x} = Ax(t), x(0) = x_0$$

is the function $x(t) = e^{At}x_0$ [5]. The expression e^{At} is the *matrix exponential* function. Examples of such equations arise in control theory and tracking applications.

A key application is the tracking of a ballistic target using noisy measurements. In this case, the matrix A above is actually a non-linear function of both x and t. The extended Kalman filter (EKF) has been used in these tracking applications [1, 2]. The typical formulation of the EKF uses a first or second-order approximation to the solution of the differential equation to save operations [3]. While such implementation is efficient, it has been shown that in some conditions the EKF may show significant bias in altitude and ballistic coefficient [6]. Under such conditions it may be preferable to use the matrix exponential function directly.

In this paper we describe and benchmark an implementation of the matrix exponential function. The implementation is based on the standard technique of "scaling and squaring" from the literature [4, 5]. The major kernels in this technique are matrix multiplication and Gaussian elimination. In the matrix multiply kernel, the implementation makes use of SIMD vector extensions present on the PowerPC G4 (Altivec) and the Intel Xeon (SSE-2). Although the use of the matrix exponential expands the operation count of the extended Kalman filter substantially, benchmarks of the implementation show that the workload is well within the capabilities of modern processors.

References

[1] Michael Athans, Robert H. Whiting, and Michael Gruber. A suboptimal estimation algorithm with probabilistic editing for false measurements with applications to target tracking with wake phenomena. *IEEE Transactions on Automatic Control*, 22(3):372–384, June 1977.

[2] Y. Bar-Shalom, X. Li, and T. Kirubarajan. *Estimation with Applications to Tracking and Navigation.* John Wiley and Sons, 2001.

[3] A. Farina, B. Ristic, and D. Benvenuti. Tracking a ballistic target: comparison of several nonlinear filters. *IEEE Transactions on Aerospace and Electronic Systems*, 38(3):854–867, July 2002.

[4] Gene H. Golub and Charles F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, 3rd edition, 1996.

[5] Cleve Moler and Charles Van Loan. Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later. *SIAM Review*, 45(1):3–49, March 2003.

[6] B. Ristic, A. Farina, D. Benvenuti and M. S. Arulampalam. Performance bounds and comparison of nonlinear filters for tracking a ballistic object on re-entry. *IEE Proceedings on Radar and Sonar Navigation*, 150(2):65–70, April 2003.

*This work sponsored by the Department of the Navy under Air Force Contract F19628-00-C-0002. Opinions, interpretations, conclusions, and recommendations are those of the authors and are not necessarily endorsed by the United States Government.

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE **01 FEB 2005**	2. REPORT TYPE **N/A**	3. DATES COVERED **-**

4. TITLE AND SUBTITLE **Implementing the Matrix Exponential Function on Embedded Processors**	5a. CONTRACT NUMBER
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **MIT Lincoln Laboratory**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release, distribution unlimited

13. SUPPLEMENTARY NOTES
See also ADM00001742, HPEC-7 Volume 1, Proceedings of the Eighth Annual High Performance Embedded Computing (HPEC) Workshops, 28-30 September 2004 Volume 1., The original document contains color images.

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**UU**	**16**	

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18

Implementing the Matrix Exponential Function on Embedded Processors

James Lebak

Andrea Wadell

Massachusetts Institute of Technology
Lincoln Laboratory

Eighth Annual High-Performance Embedded Computing Workshop (HPEC 2004)
30 Sep 2004

This work is sponsored by the United States Navy under Air Force Contract F19628-00-C-0002. Opinions, interpretations, conclusions, and recommendations are those of the authors and are not necessarily endorsed by the United States Government.

Definition
[Moler and Van Loan,2003]

The solution to the differential equation

$$\dot{x} = Ax(t)$$

$$x(0) = x_0$$

is given by

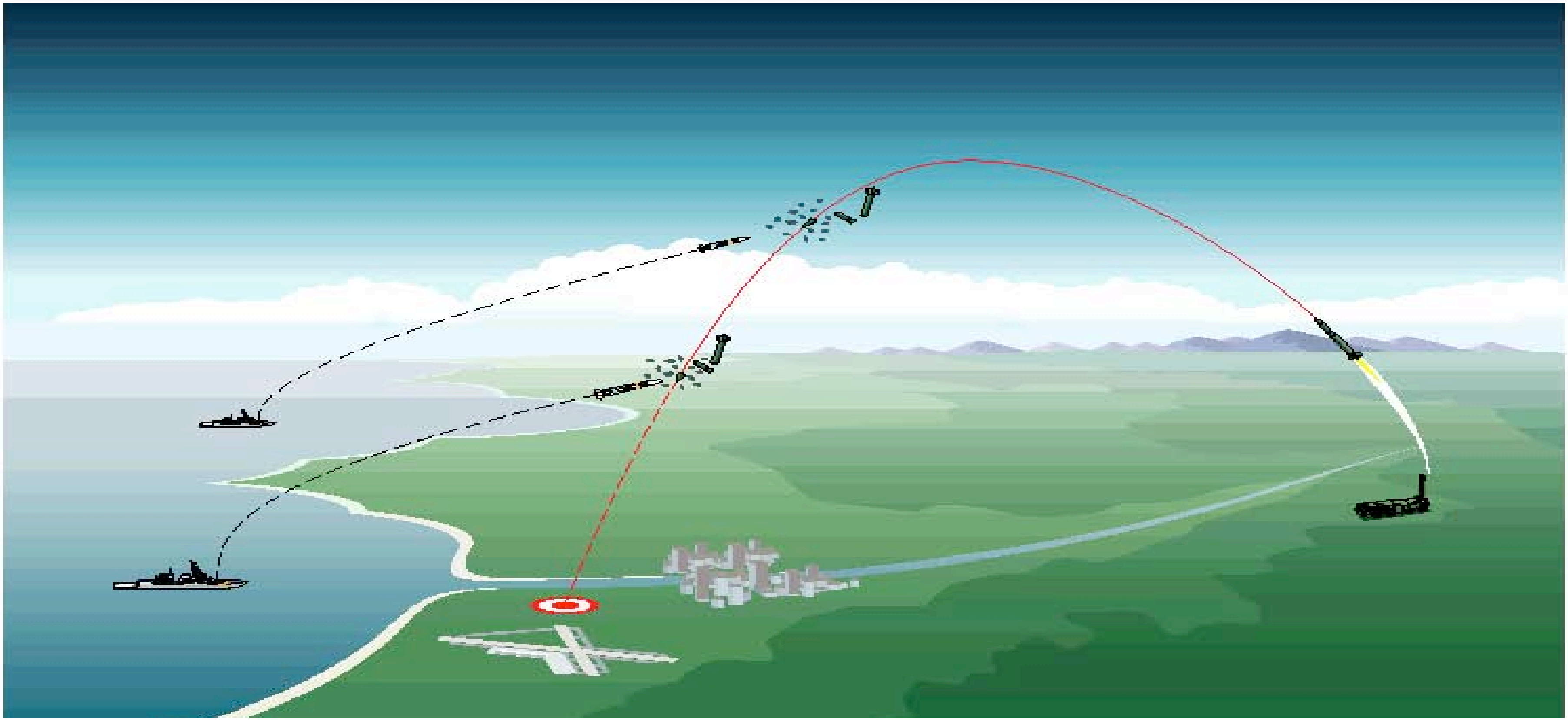
$$x(t) = e^{At} x_0$$

Where e^{At} is the matrix exponential function,

$$e^{At} = I + At + \frac{A^2 t^2}{2!} + \dots$$

Notice that if $A = [a_{ij}], e^{At} \neq [e^{a_{ij} t}]$ in general.

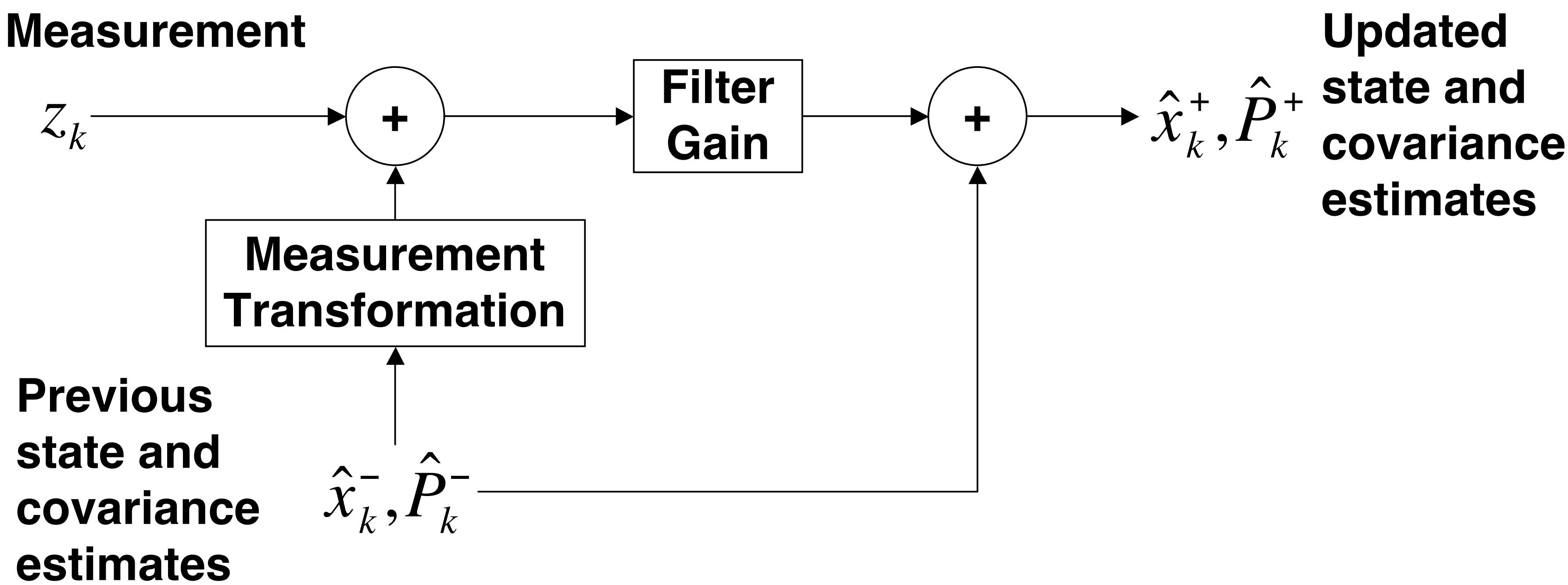
Application: Ballistic Target Tracking



- **Tracking of a ballistic target using noisy measurements**
- **Tracking accomplished using the *extended Kalman filter***
  - **"extended" means that system dynamics are non-linear**

The Extended Kalman Filter

Estimate next state based on previous state and new measurement



$$\hat{P}_k^+ = \Phi \hat{P}_k^- \Phi + Q(t),$$

where $\Phi = e^{Jt}$ **for a matrix** J**,**

and $Q(t)$ **is the process noise covariance.**

Calculation Overview

Preferred method, Padé approximation, is only valid when $||A||$ is small

Use the fact that $e^A = (e^{A/m})^m$

1. **Choose an integer j and scale A by $m=2^j$**
2. **Use a Padé approximation to calculate** $E = e^{A/2^j}$
3. **Perform j matrix multiplies to calculate** E^{2^j}

This technique is referred to as "scaling and squaring" [4,5].

Padé Iteration Algorithm

```
X = A;
c = 1;
E = I;
D = I;
for(k = 1; k <= q; k++) // q=number of iterations
{
   c = c * (q-k+1) / (k*(2*q-k+1));
   X = A*X;                  // Matrix multiply
   E = E + cX;               // Matrix scale and add
   if (k is even)            // Matrix add or subtract
     D = D + cX;
   else
     D = D - cX;
}
E = D\E;                     // Solve using LU factorization
```

Implementation Overview

Step	Operations	Percentage of op count
Scale the matrix A	Elementwise multiply	<2%
Padé iteration	Matrix multiply, scale, add	50-75%
LU and backsolve		3-6%
Repeated squaring	Matrix multiply	13-50%

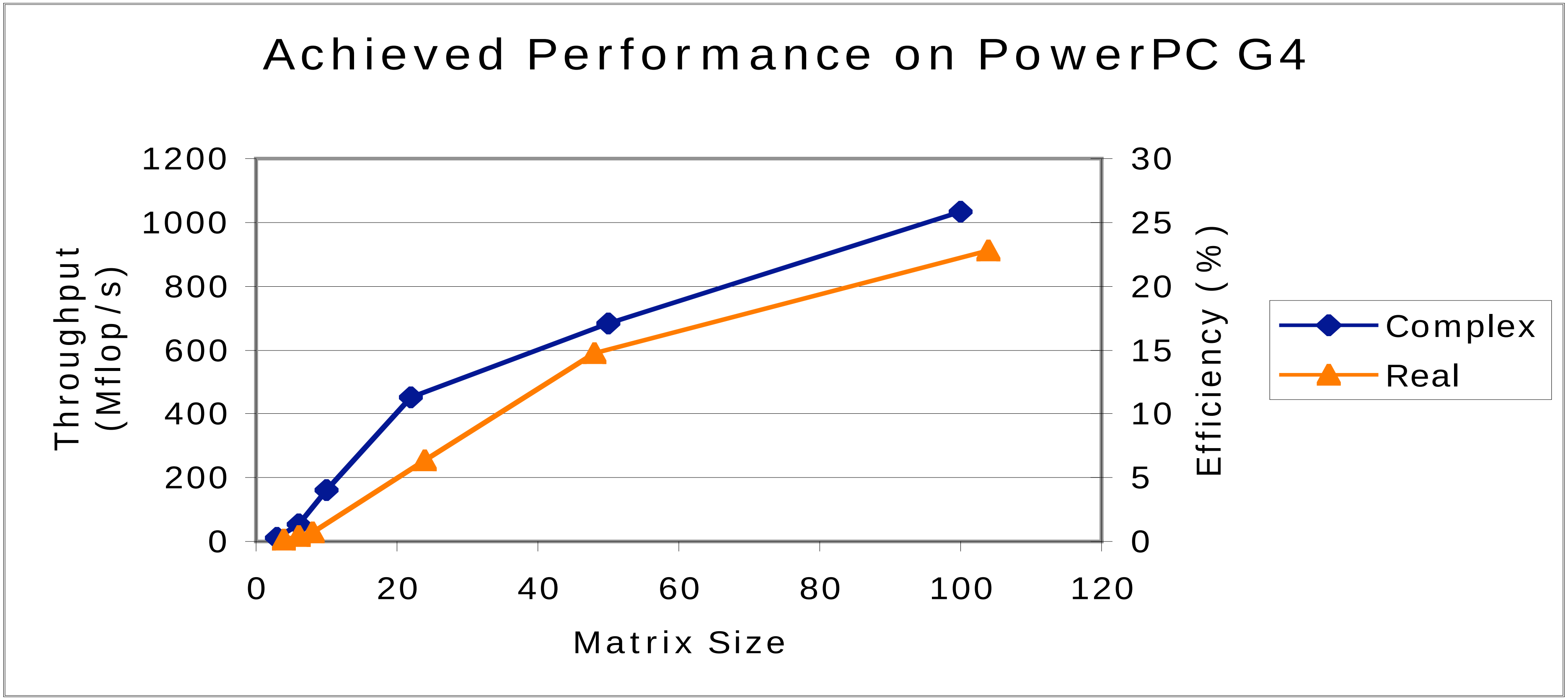
Op counts assume 6 Padé iterations

Implementation Features

- Single-precision real or complex float
- C++
- Uses an object for storage
- Calls VSIPL routines
- Uses Altivec-optimized matrix multiply
- Choose accuracy to match limits of single-precision calculations

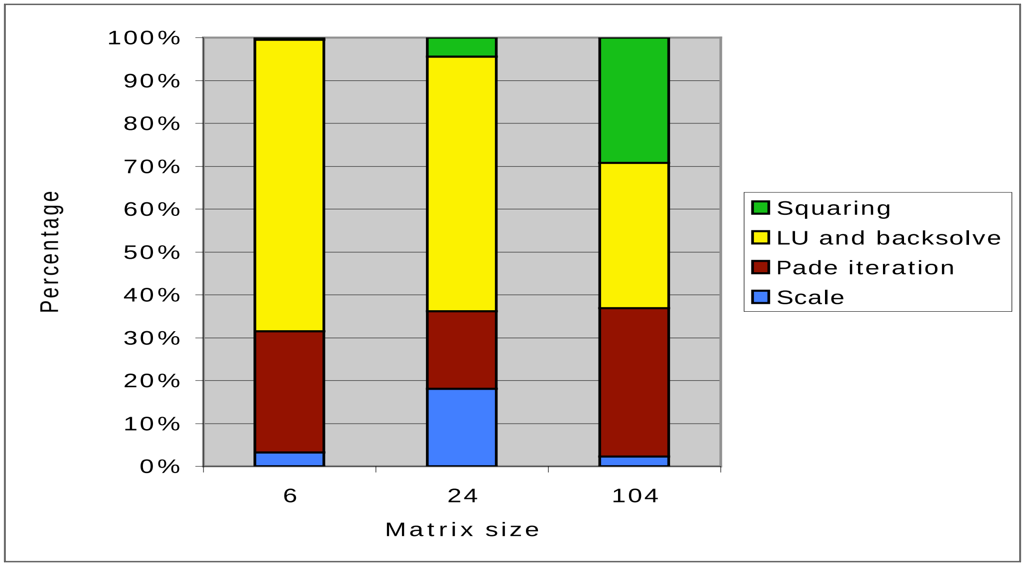
```
void create(Matrix<T> &A,          // Allocates memory & initializes
            Matrix<T> &E);         // LU factorization
void run(   Matrix<T> &A,          // Performs computation
            Matrix<T> &E);
```

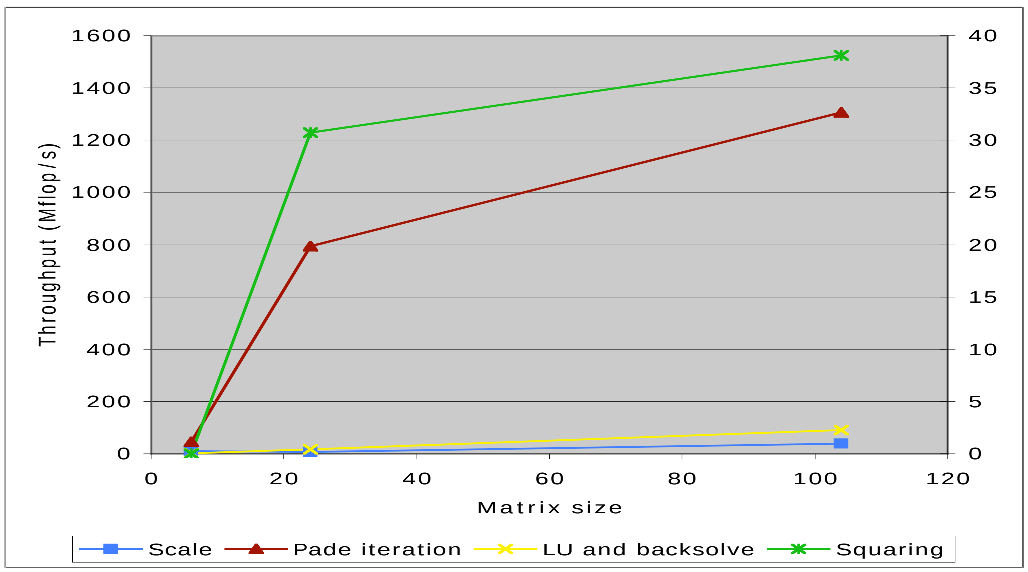
Performance



- **Platform: Mercury 500 MHz PowerPC G4**
- **Achieves respectable performance for large matrices**
- **For tracking, sizes of interest are small — 6x6 matrices**
  - **A tuned implementation could be produced for this size**

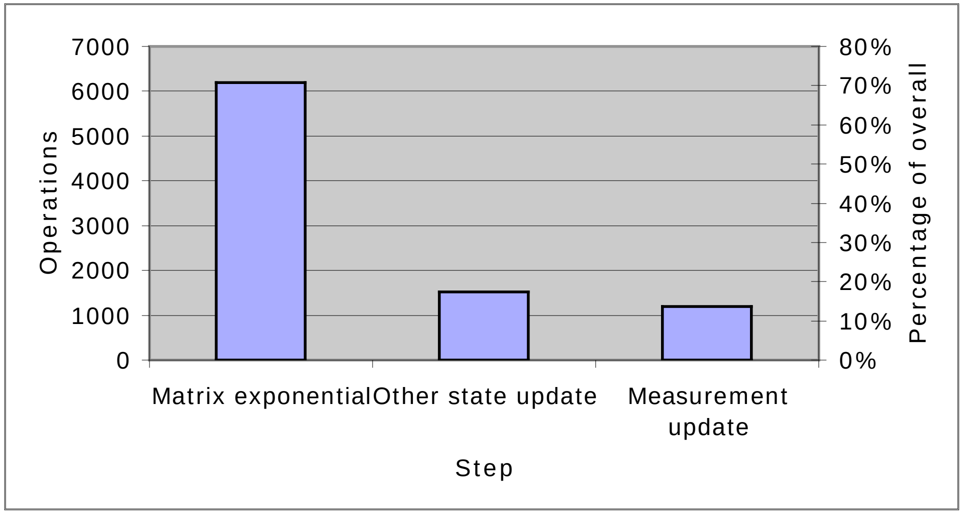
Performance Breakdown

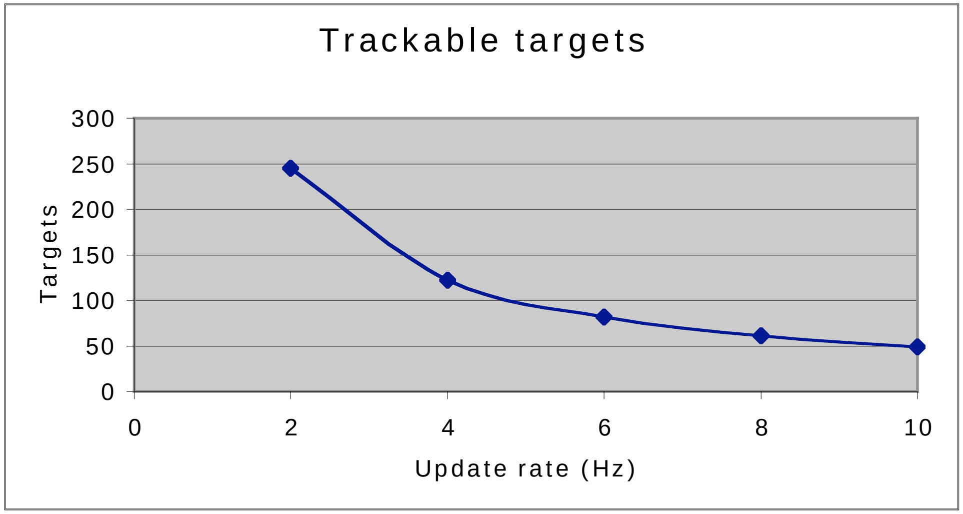




- Performance breakdown on PowerPC G4
- Steps based on matrix multiply are more efficient than other steps
- For large matrices, matrix multiply steps still consume most of the execution time
- LU/backsolve is a substantial percentage of time despite being a low percentage of the op count

The Matrix Exponential in Tracking





- **Matrix exponential is a substantial part of the EKF's operation count**
- **How many targets could a single processor track?**
  - **Assume 500 MHz PPC G4**
  - **Use execution time of 6x6 real matrix exponential**
  - **Assume remainder of EKF has efficiency comparable to LU factorization (~0.04%)**
  - **Vary track rate from 2-10 Hz**
- **A single processor can potentially track many targets**

Conclusions

- Matrix exponential function is important for tracking applications
- A large percentage of the operations are matrix multiply functions
- An efficient implementation of this function allows it to be used in an extended Kalman filter
- Many targets can be tracked using even a single processor
  - Using multiple processors obviously allows more targets to be tracked

References

[1] Michael Athans, Robert H. Whiting, and Michael Gruber. A suboptimal estimation algorithm with probabilistic editing for false measurements with applications to target tracking with wake phenomena. *IEEE Transactions on Automatic Control*, 22(3):372–384, June 1977.

[2] Y. Bar-Shalom, X. Li, and T. Kirubarajan. *Estimation with Applications to Tracking and Navigation*. John Wiley and Sons, 2001.

[3] A. Farina, B. Ristic, and D. Benvenuti. Tracking a ballistic target: comparison of several nonlinear filters. *IEEE Transactions on Aerospace and Electronic Systems*, 38(3):854–867, July 2002.

[4] Gene H. Golub and Charles F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, 3rd edition, 1996.

[5] Cleve Moler and Charles Van Loan. Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later. *SIAM Review*, 45(1):3–49,March 2003.

[6] B. Ristic, A. Farina, D. Benvenuti and M. S. Arulampalam. Performance bounds and comparison of nonlinear filters for tracking a ballistic object on re-entry. *IEE Proceedings on Radar and Sonar Navigation*, 150(2):65–70, April 2003.

Implementing the Matrix Exponential Function on Embedded Processors

James Lebak

Andrea Wadell

Massachusetts Institute of Technology

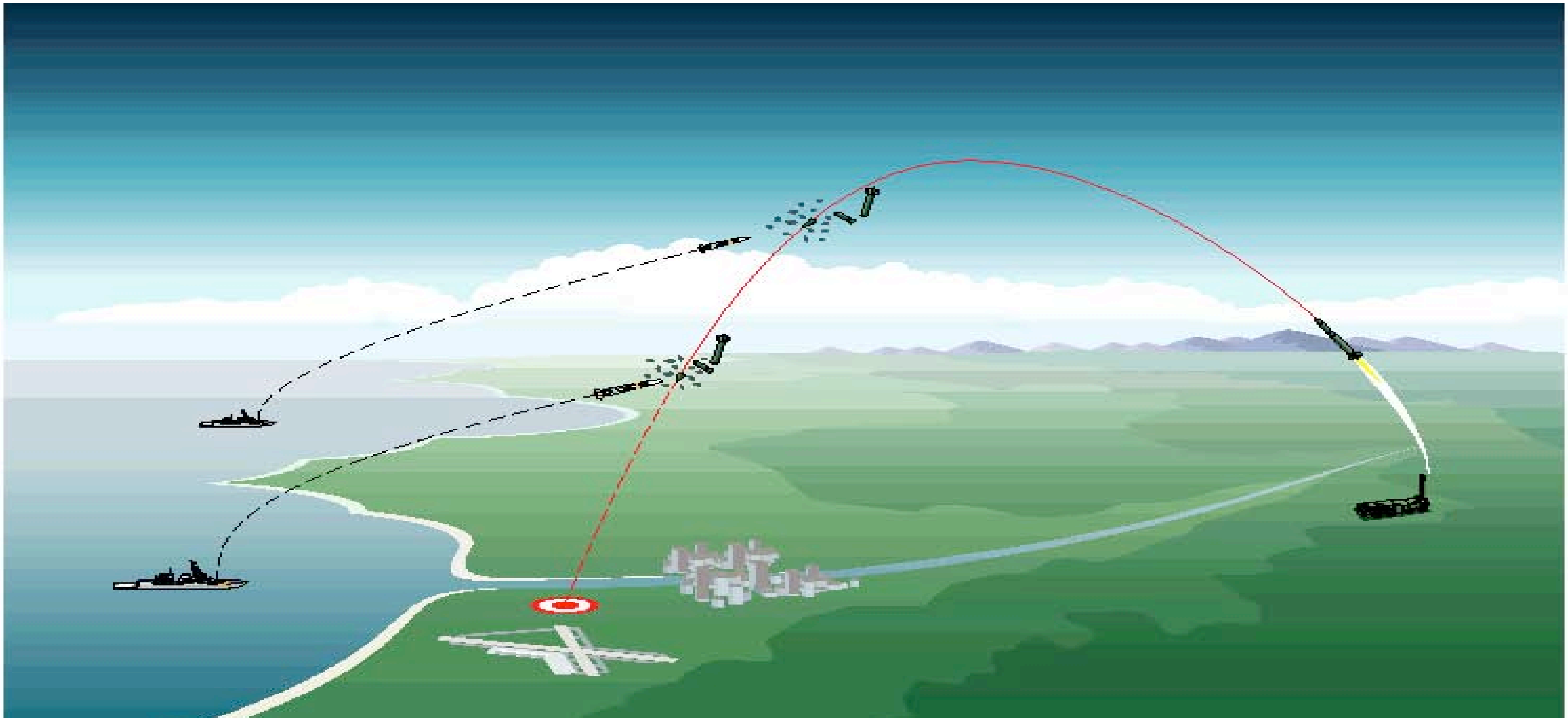
Lincoln Laboratory

Eighth Annual High-Performance Embedded Computing Workshop (HPEC 2004)

30 Sep 2004

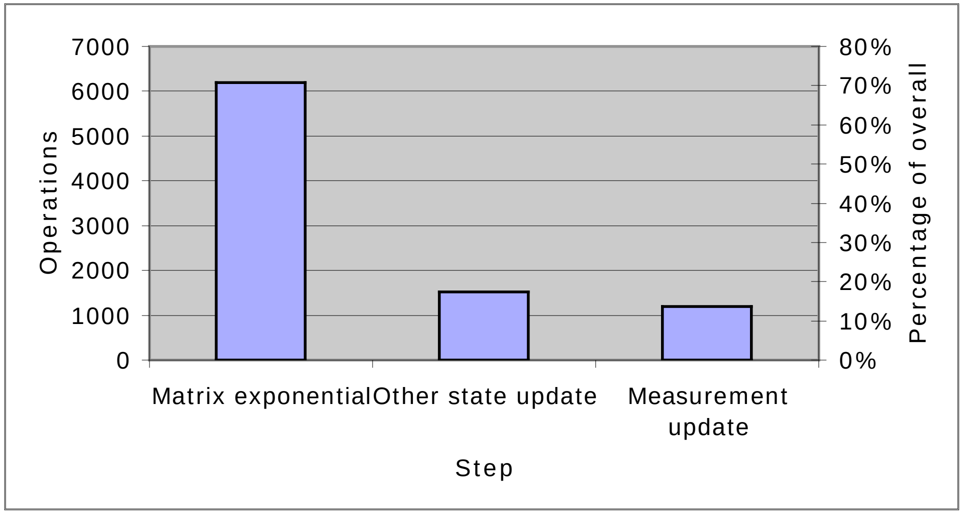
This work is sponsored by the United States Navy under Air Force Contract F19628-00-C-0002. Opinions, interpretations, conclusions, and recommendations are those of the authors and are not necessarily endorsed by the United States Government.

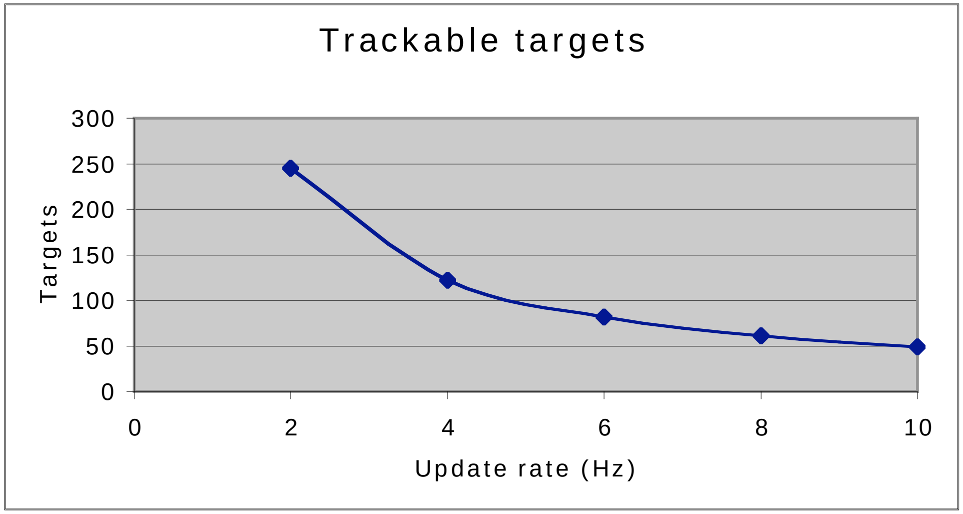
Application: Ballistic Target Tracking



- **Tracking of a ballistic target using noisy measurements**
- **Tracking accomplished using the *extended Kalman filter***
  - **"extended" means that system dynamics are non-linear**

The Matrix Exponential in Tracking





- **Matrix exponential is a substantial part of the EKF's operation count**
- **How many targets could a single processor track?**
  - **Assume 500 MHz PPC G4**
  - **Use execution time of 6x6 real matrix exponential**
  - **Assume remainder of EKF has efficiency comparable to LU factorization (~0.04%)**
  - **Vary track rate from 2-10 Hz**
- **A single processor can potentially track many targets**