

# Simulação de um sistema de controlo de tiro de um carro de combate utilizando o *kernel* de tempo real S.Ha.R.K.

André Carvalho, Paulo Felício

**Resumo –** Este artigo descreve o trabalho desenvolvido no âmbito da disciplina de opção de Sistemas de Tempo-Real, com o objectivo de criar um simulador, num ambiente multi-tarefa com garantias temporais, de um sistema de controlo de tiro de um carro de combate, ou seja, todo o sistema necessário para que um carro de combate realize a aquisição, seguimento e eventual destruição de um alvo que esteja dentro da sua área de “visão”.

Inicialmente descreve-se de forma sumária o sistema real a simular, procurando extrair os módulos que o compõem. No seguimento da referida análise é efectuada a transposição para o motor de simulação, terminando com uma breve alusão à mecânica de utilização do executivo de tempo-real SHaRK (Soft and Hard Real-time Kernel).

## I. INTRODUÇÃO

Ao implementar um simulador, temos de ter em conta a forma como os diferentes elementos do sistema se comportam e interagem uns com os outros, bem como a forma como se processa a ‘passagem do tempo’. Este ‘tempo’ pode não ser o tempo tal como o nosso relógio de pulso o marca, mas sim a referência que vamos impor ao simulador, comum a todos os elementos do processo e dependente do ritmo de evolução do mesmo. Definimos assim o Tempo-Real do sistema.

Cada elemento que compõe o sistema tem um comportamento associado, que é descrito por um modelo matemático, e condicionado por factores do ambiente em que está inserido. Estes modelos são depois alvo de simplificações ponderadas, de forma a facilitar a implementação e a tornar a aplicação final menos ‘pesada’.

Neste artigo apresentamos uma simulação de um sistema de controlo de tiro de um carro de combate. Começamos por descrever de forma sumária o sistema real a simular, procurando extrair os módulos que o compõem. Seguidamente faz-se a transposição destes módulos para o motor de simulação, terminando com uma breve alusão à mecânica de utilização do executivo de tempo-real SHaRK (Soft and Hard Real-time Kernel).

## II. ASPECTOS ESPECÍFICOS DO SIMULADOR

### Sistema a simular

Um sistema de controlo de tiro pode ser, na sua essência de sistema computacional, dividido em três componentes

base (Fig.1): os sensores de ambiente, os actuadores mecânicos e o computador de balística.

Na nossa representação do sistema consideramos a existência de dois sensores de ambiente: um visor com

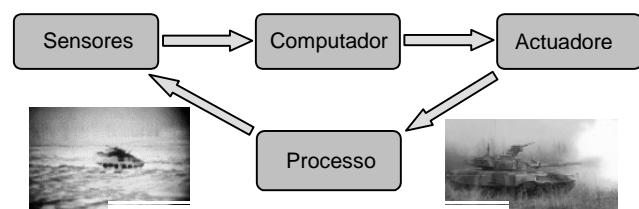


Fig.1 – Elementos do sistema a simular

capacidade sensorial térmica (de rotação livre – 360°) e um telémetro *laser*, integrado no visor rotativo. No campo dos actuadores, incluímos um actuador de rotação da torre do carro de combate, outro para a rotação do visor térmico e um terceiro destinado à elevação do canhão. Por último, as funcionalidades do computador de balística foram implementadas de forma distribuída, nomeadamente nas tarefas de controlo do movimento do visor térmico e da torre.

Para possibilitar o teste do sistema introduzimos alguns elementos externos, nomeadamente um campo de batalha virtual e um alvo com uma elevada temperatura associada (assinatura térmica), factor que permite ao visor do carro de combate detectar a sua presença. Por último, foi modelada a presença de um projéctil, indispensável para a validação da simulação.

### Modelação do sistema

O sistema em análise, o carro de combate, foi dividido em várias partes, cada uma delas com uma funcionalidade específica mas dependendo entre si, formando assim uma estrutura hierárquica (Fig.2).

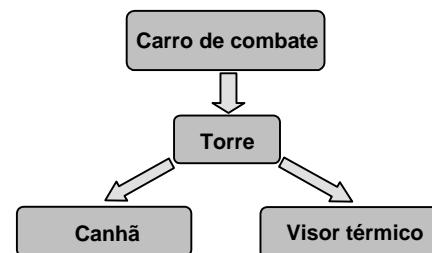


Fig.2 – Hierarquia do carro de combate

A cada elemento que compõe o sistema a simular é associada uma tarefa específica, destinada a simular o processo físico descrito pelo modelo desse elemento. Por exemplo, um sensor térmico real gera uma imagem recolhendo informação das radiações infra-vermelhas emitidas pelos objectos. No simulador, essa informação é transmitida ao elemento sensor térmico através de um gestor de ambiente, uma tarefa à qual é permitido aceder a toda a informação do meio. Desta forma, esta relaciona a orientação do sensor com a posição do alvo, verifica se este está dentro do cone de visão do sensor e devolve ao sensor uma matriz representativa do mapa de temperaturas visto pelo sensor e, caso esteja no campo de visão, a distância ao alvo (simulando o uso do telémetro). A matriz obtida tem uma dimensão definida pela abertura óptica do sensor e resolução angular do actuador do sensor. Cabe ainda ao gestor de ambiente detectar os impactos dos projéctiles com o alvo, actualizando o estado do mesmo.

A obtenção da direcção do alvo é realizada durante a rotação do sensor pelo varrimento da matriz em busca de uma posição com temperatura elevada. Caso seja encontrada, é definido um arco diferencial para o centro da matriz, conduzindo a um movimento do sensor visando a anulação dessa diferença, acompanhado de uma rotação da torre para uma orientação adequada. O projéctil é depois disparado caso se verifiquem uma série de condições (alvo adquirido, alvo dentro do alcance da arma e torre correctamente apontada). Para melhorar o seguimento, foi introduzido no controlo do movimento do sensor um compensador do tipo PI, activado quando este consegue visualizar o alvo. Em caso de perda do seguimento, o sensor regressa ao modo de busca.

As relações entre as diversas tarefas, assim como alguns dos parâmetros fundamentais dos elementos do sistema podem ser observados na figura 3. Dependendo do grau de realismo que se pretenda introduzir no simulador, poderão, no futuro, ser adicionados mais parâmetros como por exemplo as condições atmosféricas.

#### *Utilização do kernel S.Ha.R.K*

O kernel S.Ha.R.K. foi projectado para aplicações *hard*, *soft* e *non real-time*, sendo totalmente configurável, disponibilizando variados módulos de escalonamento de tarefas e protocolos de acesso a recursos partilhados. A gestão das tarefas apoia-se no conceito de modelo de tarefa, ao qual ficam associadas as características temporais da tarefa, como a periodicidade, o tempo de execução máximo, a imposição de garantias de cumprimento de prazos na sua execução. No acto de criação de cada tarefa é associado a esta um dos modelos definidos cujos parâmetros ficam guardados no seu TCB (Task Control Block), ficando a tarefa pronta a ser activada.

O simulador implementado atribuiu a cada elemento do sistema não apenas as tarefas supra referidas, que por traduzirem a evolução do sistema foram modeladas como *hard real-time* (de forma a garantir a sua execução em instantes periódicos bem definidos), mas também uma

tarefa periódica não crítica (*soft real-time*) responsável pela representação gráfica do respectivo elemento.

No kernel S.Ha.R.K o código das tarefas é implementado recorrendo a um ciclo infinito em que cada repetição corresponde a uma instância da tarefa. No final do ciclo, isto é de cada instância, é invocada uma primitiva que coloca a tarefa em *sleep* até nova activação. Para escalonar as tarefas do tipo *hard* usámos o algoritmo EDF (*earliest deadline first*) nativo do S.Ha.R.K, o qual atribui prioridade máxima às tarefas com *deadline* mais próxima. Caso uma tarefa não cumpra o *wcet* (*worst case execution time*) declarado, ou viole a respectiva *deadline*, é levantada uma excepção.

A simulação de diversas entidades lógicas (tarefas) num só processador obriga a que exista uma partilha do tempo de processador entre elas que satisfaça as necessidades de cada uma. Para determinar as necessidades das tarefas, os respectivos períodos foram definidos tendo em conta a resolução dos parâmetros do sistema físico a simular, utilizando o intervalo mínimo em que é observável uma alteração do estado de um elemento, quando este varia ao seu ritmo máximo.

Para obter uma contagem do tempo na ordem dos  $\mu$ s, foi

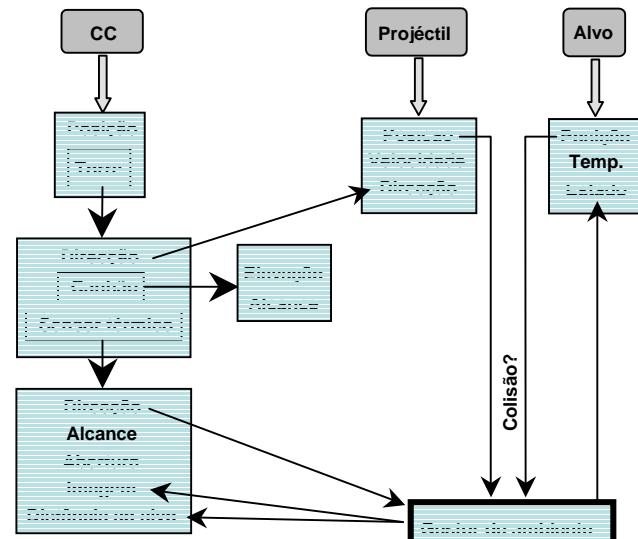


Fig. 3 – Interacção entre as diferentes tarefas

utilizado o método de sincronização ‘One Shot’, que utiliza o timer de elevada resolução dos processadores Pentium, o *Time Stamp Counter*, e permite uma introdução directa e calibrada dos períodos independentemente do CPU onde a simulação é executadas (relógio de referência).

#### III. CONCLUSÕES

Neste artigo descreveu-se de forma sucinta a simulação de um sistema de controlo de tiro de um carro de combate, utilizando o kernel de tempo-real S.Ha.R.K. Este trabalho foi desenvolvido no âmbito da disciplina de Sistemas de Tempo-Real, opção do 5º ano da LEET e da LECT.

Abordou-se o modelo físico subjacente ao simulador, bem como o encapsulamento dos elementos do simulador em tarefas, e suas inter-relações.

Em termos de desempenho, verificou-se que o peso computacional do simulador construído é relativamente baixo, sendo a parte mais significativa a da apresentação do processo ao utilizador. Este aspecto pode ser minorado reduzindo adequadamente a taxa de refreshamento da interface gráfica.

O kernel utilizado apresenta-se como apropriado a este tipo de aplicações pois, para além da utilização de um ambiente de programação do tipo *multi-tasking* que facilita o encapsulamento e isolamento dos vários elementos do simulador, permite associar restrições temporais às várias tarefas, favorecendo a obtenção de um comportamento temporal coerente e, como tal, mais fiel ao sistema original.

#### BIBLIOGRAFIA

- [1] Gai, Paolo, "S.Ha.R.K. User Manual", <http://shark.sssup.it>

- [2] Buttazzo, Giorgio C., "Hard Real-time Computing Systems – Predictable Scheduling Algorithms and Applications", Kluwer Academic Publishers, 1997
- [3] Almeida, Luís, Apontamentos de Sistemas de Tempo-Real, 2003, DETUA  
<http://sweet.ua.pt/~lda/str/str.htm>
- [4] Carvalho, Rómulo, "Física", 7<sup>a</sup>Ed., 1994, Livraria Sá da Costa Ed.
- [5] Smoothey, Craig O., "The efficiency of the Runge-Kutta and Shooting Methods as an aiming system for the M829 Anti-Tank Projectile", 1999, California National University,  
<http://www.smoothey.org/mse.pdf>
- [6] Belt, Ron, "Combat vehicle visualization system", Honeywell Inc. Sensor and Guidance products,  
<http://www.cis.upenn.edu/~reich/paper11.htm>
- [7] Commander's Independent Thermal Viewer Datasheet, Raytheon Electronic Systems,  
<http://www.raytheon.com>
- [8] Canon de 120mm & Munition de 120mm pour canons OTAN, Giat Industries,  
<http://www.giat-industries.fr>
- [9] K1A1 Main Battle Tank,  
<http://www.army-technology.com>