

SOLUCIONANDO O PROBLEMA DE LATÊNCIA DAS INFORMAÇÕES NO FUTEBOL DE ROBÔS

Guilherme Luis Pauli¹, Flavio Tonidandel²

¹ Engenharia de Controle e Automação, Centro Universitário FEI

² Engenharia Elétrica, Centro Universitário FEI

{guilhermeltpauli@gmail.com, flaviot@fei.edu.br}

Resumo: Por conta da dinâmica na categoria de futebol de robôs Small Size League (SSL) o estado dos objetos em campo deve ser atualizados constantemente, entretanto a latência entre *frames* limita a performance do software. Buscando solucionar este obstáculo, foi elaborado uma rotina para realizar a atualização a cada 1ms que consiste no uso do modelo dinâmico dos objetos inseridos no filtro Kalman para preencher a brecha de informações entre pacotes, para que o software tenha sempre uma estimativa confiável da posição dos objetos em campo.

1. Introdução

A categoria de futebol de robôs Small Size League (SSL) é uma das mais dinâmicas da RoboCup, por esse motivo o robô de cada time tem de ser capaz de processar e responder aos estímulos enviados via pacotes de rádio quase instantaneamente afim de satisfazer, por exemplo, uma jogada específica ou um posicionamento no campo.

O uso de câmeras com o processamento de imagens provém informações dos robôs e da bola em campo como, por exemplo, a localização geométrica do seu centro e, para os robôs, sua orientação. Um problema comum quando se lida com múltiplas câmeras é o aparecimento de um efeito chamado de *overlapping* que é um mesmo espaço sendo visto por diferentes câmeras e apresentando regiões geométricas distintas gerando posições diferentes para um mesmo objeto que esteja naquela região. Cada time deve ter seu sistema capaz de receber os pacotes de informações e conseguirem lidar com qualquer tipo de problema de Sistemas Operacionais em Tempo Real (RTOS, em inglês), como a latência e informações ruidosas [1].

Esta pesquisa de iniciação científica tem seu foco em corrigir problemas que ocorrem em paralelo aos jogos, sendo eles, o lapso e a latência de informações gerados pelo sistema de visão junto com o problema de *overlap* e que afetam o desempenho da equipe RoboFEI. Pensando na dinâmica desta categoria é muito importante que a atualização da posição dos objetos seja confiável e contínua, entretanto o sistema de visão (que fornece tais informações) tem uma dependência, que é a capacidade da câmera em gerar um novo *frame* quadro tornando a taxa de atualização dependente da taxa de quadros da câmera já que essa frequência é muito menor comparada com a frequência do processador e performance do software de qualquer equipe. Para solucionar os problemas apresentados será implementado uma associação de um método matemático chamado de filtro de Kalman junto à um con-

tador de tempo (*timer*) via software da equipe. Embora a aplicação possa servir para ambos os objetos (ou seja, para a bola e robôs) o foco de estudo e fase de testes serão feito somente para a bola.

2. Filtro de Kalman

O Filtro de Kalman Discreto (DKF, em inglês) é um recurso matemático que visa o ótimo, uma técnica de estimar estados elaborada em 1960 por Rudolf E. Kalman [2] sendo comumente utilizado em navegação robótica e sistemas dinâmicos lineares contaminados de ruído branco Gaussiano. Ele funciona através de equações estatísticas e do modelo do objeto em estudo, quantificando o erro das medições provenientes de sensores (R) e as estimadas no processo (Q), balanceando as incertezas através do ganho de Kalman. Seu funcionamento é regido por duas fases chamadas de Predição e Atualização, ou ainda *a priori* e *a posteriori*, respectivamente. Sendo, nas equações dispostas abaixo, todas as variáveis matrizes, k a referência para o estado atual e $k-1$ para o estado anterior, ou também *a priori* e *a posteriori*, respectivamente, e a notação \hat{X} representa uma estimativa.

$$\hat{X}_{k|k-1} = F\hat{X}_{k-1|k-1} + Bu_k \quad (1)$$

$$P_{k|k-1} = FP_{k-1|k-1}F^T + Q \quad (2)$$

$$Y_{k|k} = Z_{k|k} - H\hat{X}_{k|k-1} \quad (3)$$

$$K_{k|k} = \frac{P_{k|k-1}H^T}{HP_{k|k-1}H^T + R} \quad (4)$$

$$\hat{X}_{k|k} = \hat{X}_{k|k-1} + K_{k|k}Y_{k|k} \quad (5)$$

$$P_{k|k} = (I - K_{k|k}H)P_{k|k-1} \quad (6)$$

O filtro começa na fase de *Predição* estimando as variáveis de estado (\hat{X}_k) em 1 com os valores iniciais das suas variáveis e então é calculado o erro da predição dos estados (P_k) feita em 2, normalmente a matriz (P_k) é iniciada com valores na diagonal principal igual a 1. Na segunda etapa, de *Atualização*, é inserido uma nova amostra das variáveis de estado (Z_k) para que, através do ganho de Kalman (K_k), o sistema comece a convergir para o valor real. Após algumas iterações o ganho se estabiliza e atinge o ótimo para as variáveis de estado observadas. As matrizes (F), (B) e (u) são dadas pela representação em espaço de estados das eq. 7, 8, 9 e 10

3. Metodologia

Para que o DKF possa estimar as variáveis que se deseja observar é necessário extrair seu modelo dinâmico, ou seja, o modelo através de equações que descreva seu comportamento no deslocamento. Quanto mais fiel for o modelo da realidade maior será a certeza nos dados estimados pelo filtro. Como a bola tem seu comportamento linear, suas equações de dinâmica são regidas pelas equações de movimento da física e são as equações 7, 8, 9 e 10 abaixo.

$$X_k = X_{(k-1)} + \dot{X}_{(k-1)}\Delta t + \frac{1}{2}a_x\Delta t^2 \quad (7)$$

$$Y_k = Y_{(k-1)} + \dot{Y}_{(k-1)}\Delta t + \frac{1}{2}a_y\Delta t^2 \quad (8)$$

$$\dot{X}_k = \dot{X}_{(k-1)} + a_x\Delta t \quad (9)$$

$$\dot{Y}_k = \dot{Y}_{(k-1)} + a_y\Delta t \quad (10)$$

Sendo k a amostrado da variável, X e Y representando as posições em cada eixo, \dot{X} e \dot{Y} suas derivadas de primeira ordem, ou seja, velocidade em cada eixo, a_x e a_y o coeficiente de desaceleração da bola no campo e Δt a variação de tempo entre cada pacote da visão.

Para os robôs, o modelo dinâmico precisa atender à sua capacidade de movimento omnidirecional que tem comportamento não-linear, fazendo com que o DKF forneça uma estimativa não tão confiável. Seu modelo deve ser primeiro linearizado e algumas mudanças têm de ser feitas nas equações do filtro de Kalman antes de ser usado, que passa a ser chamado de Unscented Kalman Filter (UKF, em inglês) [3]. O foco da pesquisa é a bola, entretanto, seu modelo linear pode ser usado como uma aproximação da dinâmica do robô, mas apenas para pequenas distâncias e baixas variações de direção. Para situações diferentes ele pode se tornar instável.

No software da equipe RoboFEI foi utilizado um contador (*timer*) de $1ms$ (menor tempo suportado) para que a cada estouro uma rotina de atualização fosse chamada. Essa rotina utiliza o DKF com o modelo dinâmico da bola aplicado para ela e para os robôs e realiza uma estimação de suas posições dentro de campo enquanto um novo pacote de visão não está disponível.

Para determinar se o modelo é capaz de prover informações coerentes e com uma frequência maior que o tempo entre pacotes de informação, foram feitos alguns testes em que a bola foi jogada em campo e seu trajeto foi armazenado em um arquivo para ser analisado. Também foi salvo os tempos entre os pacotes da visão e o tempo entre cada atualização feita pela rotina. Para quantificar o erro presente nas informações geradas pelo DKF foi utilizado um método chamado de Erro Quadrático Relativo (RSE, em inglês). Para este estudo será tirado a sua média, passando a ser chamado de MRSE [4]. O MRSE, basicamente, quantifica (em porcentagem) o módulo do erro entre o dado simulado/estimado com a saída esperada real.

4. Resultados

Um total de 48 pontos gerados pelo DKF foram utilizados para o cálculo do erro. O método MRSE resultou em uma porcentagem de erro total de 0.1508%. Um valor muito baixo entre a predição e a linha polinomial de primeira ordem gerada para simular o comportamento da bola, confirmando o modelo dinâmico empregado. Um estudo de tempo comparando o período entre novos pacotes provenientes da visão e o período da rotina de atualização foi feito, retirando 11.000 amostras para cada comparação. O resultado mostra uma diferença notável e significativa para o mesmo número de amostras, tendo o período entre pacotes com quase 10% das amostras com período maior que sua média de $17ms$, já o período da rotina de atualização tem até 0.9% das amostras com período maior que o próprio tempo do *timer* de $1ms$. Essa comparação demonstra que, para uma mesma quantidade de amostras, a rotina traz as informações dos objetos em muito menos tempo que os pacotes do sistema de visão.

5. Conclusões

Comparando a média de tempo de ambos os métodos a diferença apresentada é significativa, sendo possível, através do *timer*, realizar até 30 atualizações antes da chegada de um novo pacote da visão. O DKF demonstrou alta confiabilidade ao gerar informações com baixo percentual de erro. Combinando o objeto *timer* com a função do DKF, pode-se dizer que é possível suprir a falta de informações no período entre pacotes, atualizando o software da equipe com informações confiáveis em quase todo instante de tempo.

6. Referências

- [1] A. Weitzenfeld, J. Biswas, M. Akar, and K. Sukvichai, "Robocup small-size league: Past, present and future," in *RoboCup 2014: Robot World Cup XVIII*, 2014.
- [2] G. Welch, G. Bishop *et al.*, "An introduction to the kalman filter," 1995.
- [3] E. A. Wan and R. Van Der Merwe, "The unscented kalman filter for nonlinear estimation," in *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No. 00EX373)*. Ieee, 2000, pp. 153–158.
- [4] O. A. Sotomayor, S. W. Park, and C. Garcia, "Multi-variable identification of an activated sludge process with subspace-based algorithms," *Control Engineering Practice*, vol. 11, no. 8, pp. 961–969, 2003.

Agradecimentos

À instituição Centro Universitário FEI e à equipe RoboFEI pela realização das medidas ou empréstimo de equipamentos para testes.

¹ Aluno de IC do Centro Universitário FEI (ou FAPESP, CNPq ou outra). Projeto com vigência de 11/18 a 10/19.