Bins = 3 ball periods

If we can add a component for avg. no. of dismissals in the key value pairs it could be beneficial

In the dictionary = ( batsman : avg runs in a bin , avg no. of dismissals in a bin , no. of bins )

( bowler : avg runs in a bin, avg no. of dismissals in a bin, no. of bins )

Batsman’s score in a match = sum [ actual runs in a bin/ avg runs in a bin ] – (no. of bins played in the game)\* 0.85 – (avg no. dismissals in a bin )

1. In case there are more than one dismissal in a bin ( as I think a 3 ball period could have 2 balls from one game and 1 ball from the next game ), then this score will include 1st ball or 2nd ball ducks which will not be the case for (no. of bins with dismissals)/(no. of bins )
2. I think even if the batsman stays not out we should keep the avg. dismissals per bin component, as it keeps some property of actual batting average. Players with higher batting avg will have this value low, also if someone is not out at the end this value will update to a lower value, so there is no harm.
3. In the sum […… ] – 0.85 \* ….. portion we are keeping the property of strike rate and also punishing players who avg high but take a lot of balls, so the other portion is fine

In case of bowlers I think we can compute the score as—

Bowler’s score in a match = sum [ no. of actual dismissals in a bin / avg no. dismissals in a bin ] – sum[ no. of actual runs in a bin/ avg. no of runs in a bin] + constant\* (no. of bins played in the game)

1. Adding runs/mean runs will actually decrease the score of a good performance, so we are subtracting it
2. Here we are placing a lot of value on wickets as wickets do turn matches around, you will notice that section will always have value more than 1
3. We need to add a constant as otherwise usually very economical bowlers can end up having a negative score if they have not taken wickets ( we can take that constant to be 1/.85 for example )

For measuring matchups, batting against a better bowler should be more rewarding

We can then update the formula as:

Batsman’s score = sum[ bowler’s avg. score before that game \* ( runs against that bowler in the game/ total runs scored)] \* previous formula

In case of bowlers having negative score batsman should not be penalized for that, so for bowlers with negative avg score we can replace them by 1 . Then playing well even against a team with all negative score bowlers will give our usual formula.

Similarly, bowlers’s score formula could be updated considering which batsman they were facing.

We would not need the additional info in the dictionary, we could get them just from in game stats.

Batsman’s Score Example:

Say, a Batsman A faced bowlers B, C, D in a game and scored 24 runs in 18 balls ( 6 bins ). Of those 24 he scored 12 off B, 6 off C and 6 off D and got out to bowler B.

Before that game: in the dictionary the batsman’s stored record is say, ( A: 4.33 , 100 ) and say 90 0f those 100 bins have dismissals.

And the Average Scores of the bowlers until that game are respectively -

B : 3.1 , C : 2 , D: -1.5

In that case the batsman’s score in that game will be ( 24/4.33 - 6\* 0.85 + 91/106 )\* ( 3.1 \* 12/24 + 2\* 6/24 + 1\* 6/24 ) = 2.99

Average Score of the batsman will then be updated taking the score of this game into account.